METAFORTH 77 - FORTH 77

FEATURES :

COMPLIES TO AND IS WRITTEN IN STANDARD FORTH 77
REF EFAUG BOOK 8

STRICT LAYOUT
EASY ACCESS TO AND CHANGE OF GENERATION PARAMETERS
COMMENTS EASY TO READ THROUGH USE OF BACKSLASH

COMPLETE GENERAL BOOTSTRAP PROCEDURES TO ACCOMODATE ROM-
BOOTSTRAP LOADER , BOOTSTRAP ASSEMBLER IS PROVIDED
--- BLOCKS 100 - 105

COMPLETE DOCUMENTATION OF FORTH OPERATING SYSTEM
DOCUMENTATION INCLUDES BLOCK INDEXING

CONSTANT AND LINK AREA ON BASE PAGE
FOR PAGE ORIENTED SYSTEMS

HOOKS FOR SECURITY ARE AVAILABLE --- BLOCK 376
HOOKS FOR MULTI-TASKER AND FOR MULTI-USER ARE AVAILABLE
MULTITASKER IMPLEMENTED --- OUTSIDE LEVEL 0

STANDARD 3-CHARACTER IDENTIFIER
2N+1 CHARACTER FORTH AND FREE OR
FLOATING CHARACTER FORTH ARE OPTIONAL

BLOCK HANDLERS ARE SEPARATED FROM DRIVERS
REF EFAUG BOOK 4 --- BLOCKS 440 610
N-BUFFER SYSTEM --- N >= 2

COMMENTS :

INSTRUCTION COUNTER IC IS VALUE ON TOP OF RETURN STACK
IN ORDER TO FACILITATE UPGRADE TO SVEND LORENSENS
MICRO-CODE FORTH

LEVEL 0 FORTH IS NEARING COMPLETION --- BLOCKS 13 - 50
PSEUDOVECTORS FOR 3 CHAR-FORTH NOT TESTED
- REMEMBER, PSEUDOVECTORS NOT COMPATIBLE
WITH OTHER-CHARACTER FORTH SYSTEMS
SECURITY IMPLEMENTED
EXTREMELY SIMPLE FILE-SYSTEM

ADIOS AMIGOS

METAFORTH 77 - FORTH 77

FEATURES :

COMPLIES TO AND IS WRITTEN IN STANDARD FORTH 77
REF EFAUG BOOK 8

STRICT LAYOUT
EASY ACCESS TO AND CHANGE OF GENERATION PARAMETERS
COMMENTS EASY TO READ THROUGH USE OF BACKSLASH

COMPLETE GENERAL BOOTSTRAP PROCEDURES TO ACCOMODATE ROM-
BOOTSTRAP LOADER , BOOTSTRAP ASSEMBLER IS PROVIDED
--- BLOCKS 100 - 105

COMPLETE DOCUMENTATION OF FORTH OPERATING SYSTEM
DOCUMENTATION INCLUDES BLOCK INDEXING

CONSTANT AND LINK AREA ON BASE PAGE
FOR PAGE ORIENTED SYSTEMS

HOOKS FOR SECURITY ARE AVAILABLE --- BLOCK 376
HOOKS FOR MULTI-TASKER AND FOR MULTI-USER ARE AVAILABLE
MULTITASKER IMPLEMENTED --- OUTSIDE LEVEL 0

STANDARD 3-CHARACTER IDENTIFIER
2N+1 CHARACTER FORTH AND FREE OR
FLOATING CHARACTER FORTH ARE OPTIONAL

BLOCK HANDLERS ARE SEPARATED FROM DRIVERS
REF EFAUG BOOK 4 --- BLOCKS 440 610
N-BUFFER SYSTEM --- N >= 2

COMMENTS :

INSTRUCTION COUNTER IC IS VALUE ON TOP OF RETURN STACK
IN ORDER TO FACILITATE UPGRADE TO SVEND LORENSENS
MICRO-CODE FORTH

LEVEL 0 FORTH IS NEARING COMPLETION --- BLOCKS 13 - 50
PSEUDOVECTORS FOR 3 CHAR-FORTH NOT TESTED
- REMEMBER, PSEUDOVECTORS NOT COMPATIBLE
WITH OTHER-CHARACTER FORTH SYSTEMS
SECURITY IMPLEMENTED
EXTREMELY SIMPLE FILE-SYSTEM

ADIOS AMIGOS

UTRECHT 771205 HANS JAN FRANS THEO

```
   13  ( BASIC FORTH ASSEMBLER FOR HP 2100          JV/780313   1 OF ** )
2  14  ( BASIC FORTH ASSEMBLER FOR HP 2100          JV/780330   2 OF ** )
   15  ( BASIC FORTH DEFINITIONS                    HN/780310   3 OF ** )
4  16  ( BASIC FORTH DEFINITIONS                    HN/780208   4 OF ** )
   17  ( BASIC FORTH DEFINITIONS                    HN/780310   5 OF ** )
6  18  ( BASIC FORTH STATE AND PRECEDENCE           HN/780208   6 OF ** )
   19  ( BASIC FORTH                                JV/780120   7 OF ** )
8  20  ( BASIC FORTH                                JV/780313   8 OF ** )
   21  ( BASIC FORTH ASSEMBLER MACRO'S              JV/780330   9 OF ** )
10 22  ( BASIC FORTH COMPILER DIRECTIVES            JV/780330  10 OF ** )
   23  ( BASIC FORTH COMPILER DIRECTIVES            JV/780119  11 OF ** )
12 24  ( BASIC FORTH COMPILER DIRECTIVES            HN/780310  12 OF ** )
   25  ( BASIC FORTH COMPARE OPERATORS              JV/780120  13 OF ** )
14 26  ( BASIC FORTH CONDITIONAL INTERPRETATION     JV/771201  14 OF ** )
   27  ( BASIC FORTH FIX COMPUTER TYPE              HN/780310  15 OF ** )
16 28  ( BASIC FORTH FIX COMPUTER TYPE              JV/780201  16 OF ** )
   29  ( BASIC FORTH ARITHMETIC                     JV/771201  17 OF ** )
18 30  ( BASIC FORTH VOCABULARY HANDLING           HN/780310  18 OF ** )
   31  ( BASIC FORTH VOCABULARY HANDLING           JV/780330  19 OF ** )
20 32  ( BASIC FORTH VOCABULARY HANDLING           JV/780118  20 OF ** )
   33  ( BASIC FORTH                                JV/780313  21 OF ** )
22 34  ( BASIC FORTH                                JV/780119  22 OF ** )
   35  ( BASIC FORTH                                HN/780119  23 OF ** )
24 36  ( BASIC FORTH                                HN/780212  24 OF ** )
   37  ( BASIC FORTH                                HN/780212  25 OF ** )
26 38  ( BASIC FORTH                                JV/780330  26 OF ** )
   39  ( BASIC FORTH                                JV/780119  27 OF ** )
28 40  ( BASIC FORTH                                HN/780310  28 OF ** )
   41  ( BASIC FORTH                                JV/780330  29 OF ** )
30 42  ( BASIC FORTH                                HN/780310  30 OF ** )
   43  ( BASIC FORTH ERROR HANDLING                 JV/780316  31 OF ** )
32 44  ( BASIC FORTH                                JV/780125  32 OF ** )
   45  ( BASIC FORTH                                HN/780212  33 OF ** )
34 46  ( BASIC FORTH                                JV/780316  34 OF ** )
   47  ( BASIC FORTH                                JV/780403  35 OF ** )
36 80  ( RESIDENT BLOCKS                           JV/780403   1 OF ** )
   81  ( RESIDENT BLOCKS                           JV/780403   2 OF ** )
38 90  ( FORTH [ ERROR ] MESSAGES   1 - 15         JV/780120   1 OF  2 )
   91  ( FORTH [ ERROR ] MESSAGES  16 - 30         JV/780403   2 OF  2 )
40 97  ( TEMPORARY CONVERSION TO STANDARD FORTH    JV/780321   1 OF  3 )
   98  ( TEMPORARY CONVERSION TO STANDARD FORTH    JV/780330   2 OF  3 )
42 99  ( TEMPORARY CONVERSION TO STANDARD FORTH    JV/780315   3 OF  3 )
   100  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           JV/770926   1 OF  6 )
44 101  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           JV/770926   2 OF  6 )
   102  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           JV/770926   3 OF  6 )
46 103  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           JV/770926   4 OF  6 )
   104  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           JV/770926   5 OF  6 )
48 105  ( BLOCK ASSEMBLER FOR BOOTSTRAPS           HN/780310   6 OF  6 )
   110  ( ASSEMBLER FOR HP 21MX COMPUTERS          JV/780125   1 OF  3 )
50 111  ( ASSEMBLER FOR HP 21MX COMPUTERS          JV/771010   2 OF  3 )
   112  ( ASSEMBLER FOR HP 21MX COMPUTERS          JV/771010   3 OF  3 )
52 200  ( BOOTSTRAP FOR MOST ROM LOADERS          HN/780319   1 OF  6 )
   201  ( BOOTSTRAP FOR MOST ROM LOADERS          JV/780113   2 OF  6 )
54 202  ( BOOTSTRAP FOR MOST ROM LOADERS          JV/780117   3 OF  6 )
   203  ( BOOTSTRAP FOR MOST ROM LOADERS          HN/780313   4 OF  6 )
56 204  ( BOOTSTRAP FOR MOST ROM LOADERS          HN/780313   5 OF  6 )
   205  ( BOOTSTRAP FOR MOST ROM LOADERS          HN/780313   6 OF  6 )
58 250  ( BOOTSTRAP FOR 7900 DISC ROM LOADER      JV/770926   1 OF  7 )
   251  ( BOOTSTRAP FOR 7900 DISC ROM LOADER      HN/780313   2 OF  7 )
60 252  ( BOOTSTRAP FOR 7900 DISC ROM LOADER      JV/770907   3 OF  7 )

62

64
```

```
253  ( BOOTSTRAP FOR 7900 DISC ROM LOADER          JV/770926   4 OF  7 )
254  ( BOOTSTRAP FOR 7900 DISC ROM LOADER          JV/770926   5 OF  7 )
255  ( BOOTSTRAP FOR 7900 DISC ROM LOADER          JV/770926   6 OF  7 )
256  ( BOOTSTRAP FOR 7900 DISC ROM LOADER          JV/770926   7 OF  7 )
260  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/770926   1 OF  7 )
261  ( BOOTSTRAP FOR 7900 DISC FORTH               HN/780313   2 OF  7 )
262  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/771018   3 OF  7 )
263  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/770926   4 OF  7 )
264  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/770926   5 OF  7 )
265  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/770926   6 OF  7 )
266  ( BOOTSTRAP FOR 7900 DISC FORTH               JV/770926   7 OF  7 )
300  ( METAFORTH INITIALIZATION                    JV/770926   1 OF ** )
301  ( METAFORTH INITIALIZATION                    JV/780330   2 OF ** )
302  ( METAFORTH INITIALIZATION                    JV/780313   3 OF ** )
303  ( METAFORTH INITIALIZATION                    HN/780212   4 OF ** )
304  ( METAFORTH INITIALIZATION                    JV/780403   5 OF ** )
305  ( METAFORTH INITIALIZATION                    HN/780313   6 OF ** )
310  ( METAFORTH GENERATION CONSTANTS              HN/780315   1 OF ** )
311  ( METAFORTH GENERATION CONSTANTS              HN/780313   2 OF ** )
312  ( METAFORTH GENERATION CONSTANTS              JV/780313   3 OF ** )
313  ( METAFORTH GENERATION CONSTANTS              HN/780313   4 OF ** )
314  ( METAFORTH GENERATION CONSTANTS              JV/771114   5 OF ** )
315  ( METAFORTH GENERATION CONSTANTS              HN/780212   6 OF ** )
320  ( METAFORTH BASIC GENERATION WORDS            JV/780315   1 OF ** )
321  ( METAFORTH BASIC GENERATION WORDS            JV/771018   2 OF ** )
322  ( METAFORTH BASIC GENERATION WORDS            JV/780313   3 OF ** )
323  ( METAFORTH BASIC GENERATION WORDS            JV/780321   4 OF ** )
324  ( METAFORTH BASIC GENERATION WORDS            HN/780212   5 OF ** )
325  ( METAFORTH BASIC GENERATION WORDS            JV/780118   6 OF ** )
326  ( METAFORTH BASIC GENERATION WORDS            HN/780313   7 OF ** )
327  ( METAFORTH BASIC GENERATION WORDS            HN/770929   8 OF ** )
328  ( METAFORTH BASIC GENERATION WORDS            JV/780321   9 OF ** )
330  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      HN/780319   1 OF ** )
331  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      JV/771021   2 OF ** )
332  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      JV/780330   3 OF ** )
333  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      JV/770926   4 OF ** )
334  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      JV/780330   5 OF ** )
335  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      HN/760212   6 OF ** )
336  ( METAFORTH ASSEMBLER FOR HP 2100 SERIES      HN/780313   7 OF ** )
340  ( METAFORTH GENERATION                        HN/780315   1 OF ** )
341  ( METAFORTH GENERATION                        HN/780319   2 OF ** )
345  ( METAFORTH MONITOR                           JV/771103   1 OF  1 )
350  ( METAFORTH CONSTANT AREA                     HN/780319   1 OF  1 )
355  ( METAFORTH USER AREA                         JV/780116   1 OF ** )
356  ( METAFORTH USER AREA                         HN/780313   2 OF ** )
357  ( METAFORTH USER AREA                         JV/780201   3 OF ** )
365  ( METAFORTH INTERPRETER                       HN/780319   1 OF ** )
366  ( METAFORTH INTERPRETER                       HN/780319   2 OF ** )
367  ( METAFORTH INTERPRETER                       HN/780319   3 OF ** )
366  ( METAFORTH INTERPRETER                       JV/780320   4 OF ** )
370  ( METAFORTH BASIC VOCABULARY                  HN/780313   1 OF ** )
371  ( METAFORTH BASIC VOCABULARY                  HN/780313   2 OF ** )
372  ( METAFORTH BASIC VOCABULARY                  HN/780317   3 OF ** )
373  ( METAFORTH BASIC VOCABULARY                  HN/780319   4 OF ** )
374  ( METAFORTH BASIC VOCABULARY                  HN/780313   5 OF ** )
375  ( METAFORTH BASIC VOCABULARY                  HN/780319   6 OF ** )
376  ( METAFORTH BASIC VOCABULARY                  HN/780319   7 OF ** )
380  ( METAFORTH LITERALS                          HN/780319   1 OF ** )
381  ( METAFORTH LITERALS                          HN/780319   2 OF ** )
382  ( METAFORTH LITERALS                          JV/780403   3 OF ** )
```

```
383  ( METAFORTH LITERALS                          JV/771007   4 OF ** )
390  ( METAFORTH LEVEL 0 WORDS                     JV/780320   1 OF ** )
391  ( METAFORTH LEVEL 0 WORDS                     HN/780319   2 OF ** )
392  ( METAFORTH LEVEL 0 WORDS                     HN/780319   3 OF ** )
405  ( METAFORTH RESOURCE SHARING                  HN/780319   1 OF ** )
406  ( METAFORTH RESOURCE SHARING                  JV/780321   2 OF ** )
407  ( METAFORTH RESOURCE SHARING  DMA             JV/780320   3 OF ** )
410  ( METAFORTH CHARACTER HANDLING                JV/780330   1 OF  2 )
411  ( METAFORTH CHARACTER HANDLING                JV/780330   2 OF  2 )
415  ( METAFORTH WORD HANDLING                     HN/770928   1 OF ** )
416  ( METAFORTH WORD HANDLING                     HN/780319   2 OF ** )
417  ( METAFORTH WORD HANDLING                     JV/780403   3 OF ** )
420  ( METAFORTH NUMBER CONVERSION                 JV/780315   1 OF ** )
421  ( METAFORTH NUMBER CONVERSION                 JV/780118   2 OF ** )
422  ( METAFORTH NUMBER CONVERSION                 HN/780319   3 OF ** )
423  ( METAFORTH NUMBER CONVERSION                 HN/780319   4 OF ** )
430  ( METAFORTH DICTIONARY SEARCH                 HN/780313   1 OF  7 )
431  ( METAFORTH DICTIONARY SEARCH                 HN/780313   2 OF  7 )
432  ( METAFORTH DICTIONARY SEARCH                 HN/780319   3 OF  7 )
433  ( METAFORTH DICTIONARY SEARCH                 JV/780316   4 OF  7 )
434  ( METAFORTH DICTIONARY SEARCH                 HN/780319   5 OF  7 )
435  ( METAFORTH DICTIONARY SEARCH                 HN/780208   6 OF  7 )
436  ( METAFORTH DICTIONARY SEARCH                 HN/780319   7 OF  7 )
440  ( METAFORTH BLOCK HANDLING                    JV/780201   1 OF ** )
441  ( METAFORTH BLOCK HANDLING                    JV/780313   2 OF ** )
442  ( METAFORTH BLOCK HANDLING                    JV/780330   3 OF ** )
443  ( METAFORTH BLOCK HANDLING                    HN/780319   4 OF ** )
444  ( METAFORTH BLOCK HANDLING                    HN/780319   5 OF ** )
445  ( METAFORTH BLOCK HANDLING                    HN/780319   6 OF ** )
446  ( METAFORTH BLOCK HANDLING                    JV/780330   7 OF ** )
447  ( METAFORTH BLOCK HANDLING                    JV/780330   8 OF ** )
448  ( METAFORTH BLOCK HANDLING                    JV/770929   9 OF ** )
449  ( METAFORTH BLOCK HANDLING                    HN/780319  10 OF ** )
460  ( METAFORTH BLOCK LOADING                     HN/780319   1 OF ** )
461  ( METAFORTH BLOCK LOADING                     JV/780124   2 OF ** )
462  ( METAFORTH BLOCK LOADING                     HN/780319   3 OF ** )
470  ( METAFORTH INTERPRETER INPUT                 JV/771006   1 OF ** )
475  ( METAFORTH FORTH VOCABULARY                  HN/780313   1 OF ** )
476  ( METAFORTH FORTH VOCABULARY                  HN/780319   2 OF ** )
480  ( METAFORTH ASSEMBLER VOCABULARY              HN/780313   1 OF ** )
481  ( METAFORTH ASSEMBLER VOCABULARY              HN/780319   2 OF ** )
482  ( METAFORTH ASSEMBLER VOCABULARY              JV/780320   3 OF ** )
483  ( METAFORTH ASSEMBLER VOCABULARY              JV/780117   4 OF ** )
484  ( METAFORTH ASSEMBLER VOCABULARY              HN/780319   5 OF ** )
485  ( METAFORTH ASSEMBLER VOCABULARY              HN/780319   6 OF ** )
486  ( METAFORTH ASSEMBLER VOCABULARY              JV/780320   7 OF ** )
495  ( METAFORTH GENERAL CONSOLE I/O               JV/780331   1 OF ** )
496  ( METAFORTH GENERAL CONSOLE I/O               JV/780331   2 OF ** )
500  ( METAFORTH INTERPRETER WORDS                 JV/780330   1 OF ** )
501  ( METAFORTH INTERPRETER WORDS                 HN/780319   2 OF ** )
502  ( METAFORTH INTERPRETER WORDS                 JV/780330   3 OF ** )
503  ( METAFORTH INTERPRETER WORDS                 JV/780119   4 OF ** )
505  ( METAFORTH INTERPRETER                       JV/771007   1 OF ** )
510  ( METAFORTH MISCELLANEOUS                     HN/780319   1 OF ** )
511  ( METAFORTH MISCELLANEOUS                     HN/780313   2 OF ** )
520  ( METAFORTH SOFTWARE MPY DIV                  HN/780319   1 OF ** )
521  ( METAFORTH SOFTWARE MPY DIV                  HN/780319   2 OF ** )
522  ( METAFORTH SOFTWARE MPY DIV                  HN/780319   3 OF ** )
523  ( METAFORTH SOFTWARE MPY DIV                  HN/780319   4 OF ** )
524  ( METAFORTH SOFTWARE MPY DIV                  HN/780319   5 OF ** )
```

```
525  ( METAFORTH SOFTWARE MPY DIV                    HN/780319   6 OF ** )
530  ( METAFORTH CONSOLE I/O                         HN/780319   1 OF ** )
540  ( METAFORTH MASS-STORAGE DEVICES                JV/780330   1 OF  1 )
550  ( METAFORTH RESTART                             JV/780316   1 OF ** )
551  ( METAFORTH RESTART                             JV/780320   2 OF ** )
552  ( METAFORTH RESTART                             JV/780330   3 OF ** )
553  ( METAFORTH RESTART                             JV/780330   4 OF ** )
560  ( METAFORTH POINTER INITIALIZATION              JV/780320   1 OF ** )
561  ( METAFORTH POINTER INITIALIZATION              JV/780320   2 OF ** )
562  ( METAFORTH POINTER INITIALIZATION              JV/780320   3 OF ** )
600  ( METAFORTH HP 7900A DISC I/O                   HN/780212   1 OF 12 )
601  ( METAFORTH HP 7900A DISC I/O                   HN/780313   2 OF 12 )
602  ( METAFORTH HP 7900A DISC I/O                   HN/780320   3 OF 12 )
603  ( METAFORTH HP 7900A DISC I/O                   JV/771114   4 OF 12 )
604  ( METAFORTH HP 7900A DISC I/O                   HN/780320   5 OF 12 )
605  ( METAFORTH HP 7900A DISC I/O                   JV/780330   6 OF 12 )
606  ( METAFORTH HP 7900A DISC I/O                   JV/780330   7 OF 12 )
607  ( METAFORTH HP 7900A DISC I/O                   JV/771114   8 OF 12 )
608  ( METAFORTH HP 7900A DISC I/O                   JV/780330   9 OF 12 )
609  ( METAFORTH HP 7900A DISC I/O                   JV/780330  10 OF 12 )
610  ( METAFORTH HP 7900A DISC I/O                   HN/780320  11 OF 12 )
611  ( METAFORTH HP 7900A DISC I/O                   HN/780212  12 OF 12 )
700  ( METAFORTH TTY CONSOLE I/O                     JV/780331   1 OF ** )
701  ( METAFORTH TTY CONSOLE I/O : NON-INTER         JV/780331   2 OF ** )
702  ( METAFORTH TTY CONSOLE I/O : INTERRUPT         JV/780331   3 OF ** )
703  ( METAFORTH TTY CONSOLE I/O : INTERRUPT         JV/780331   4 OF ** )
704  ( METAFORTH TTY CONSOLE I/O : INTERRUPT         JV/780331   5 OF ** )
705  ( METAFORTH TTY CONSOLE I/O                     JV/780331   6 OF ** )
706  ( METAFORTH TTY CONSOLE I/O                     JV/780331   7 OF ** )
707  ( METAFORTH TTY CONSOLE I/O                     JV/780403   8 OF ** )
708  ( METAFORTH TTY CONSOLE I/O                     JV/780331   9 OF ** )
709  ( METAFORTH TTY CONSOLE I/O                     JV/780331  10 OF ** )
1000 ( SECURITY                                      HN/780320   1 OF ** )
1001 ( SECURITY                                      HN/780313   2 OF ** )
1002 ( SECURITY                                      JV/780330   3 OF ** )
1003 ( SECURITY                                      JV/780320   4 OF ** )
1004 ( SECURITY                                      JV/780403   5 OF ** )
1005 ( SECURITY                                      JV/780320   6 OF ** )
1020 ( MONITOR MULTIPROGRAMMER                       JV/780320   1 OF ** )
1021 ( MONITOR MULTIPROGRAMMER                       HN/780315   2 OF ** )
1022 ( MONITOR MULTIPROGRAMMER                       HN/780315   3 OF ** )
1023 ( MONITOR MULTIPROGRAMMER                       HN/780315   4 OF ** )
1024 ( MONITOR MULTIPROGRAMMER                       HN/780315   5 OF ** )
1025 ( MONITOR MULTIPROGRAMMER                       HN/780320   6 OF ** )
1026 ( MONITOR MULTIPROGRAMMER OPERATOR              JV/780320   7 OF ** )
1027 ( MONITOR MULTIPROGRAMMER                       JV/780330   8 OF ** )
1028 ( MONITOR MULTIPROGRAMMER                       JV/780321   9 OF ** )
1029 ( MONITOR MULTIPROGRAMMER                       JV/780330  10 OF ** )
1030 ( MONITOR MULTIPROGRAMMER                       JV/780330  11 OF ** )
1050 ( TIME BASE GENERATOR ROUTINES                  HN/780202   1 OF  5 )
1051 ( TIME BASE GENERATOR ROUTINES                  HN/780202   2 OF  5 )
1052 ( TIME BASE GENERATOR ROUTINES                  HN/780320   3 OF  5 )
1053 ( TIME BASE GENERATOR ROUTINES                  HN/780202   4 OF  5 )
1054 ( TIME BASE GENERATOR ROUTINES                  JV/780330   5 OF  5 )
1100 ( PSEUDOVECTORS FOR 3 CHAR FORTH                HN/780320   1 OF  5 )
1101 ( PSEUDOVECTORS FOR 3 CHAR FORTH                HN/780313   2 OF  5 )
1102 ( PSEUDOVECTORS FOR 3 CHAR FORTH                HN/780313   3 OF  5 )
1103 ( PSEUDOVECTORS FOR 3 CHAR FORTH                JV/780313   4 OF  5 )
1104 ( PSEUDOVECTORS FOR 3 CHAR FORTH                JV/780330   5 OF  5 )
1110 ( FILE-HANDLING                                 HN/780313   1 OF  1 )
```

```
1120  ( EDITOR                                              JV/780330   1 OF ** )
1121  ( EDITOR                                              JV/780320   2 OF ** )
1122  ( EDITOR                                              HN/780320   3 OF ** )
1130  ( EDITOR                                              JV/780330   1 OF ** )
1131  ( EDITOR                                              JV/780330   2 OF ** )
1132  ( EDITOR                                              JV/780330   3 OF ** )
1133  ( EDITOR                                              JV/780330   4 OF ** )
1200  ( HP 7900 DISC I/O                                    HN/780313   1 OF 11 )
1201  ( HP 7900 DISC I/O                                    HN/780203   2 OF 11 )
1202  ( HP 7900 DISC I/O                                    JV/780320   3 OF 11 )
1203  ( HP 7900 DISC I/O                                    HN/780203   4 OF 11 )
1204  ( HP 7900 DISC I/O                                    HN/780203   5 OF 11 )
1205  ( HP 7900 DISC I/O                                    JV/780330   6 OF 11 )
1206  ( HP 7900 DISC I/O                                    JV/780330   7 OF 11 )
1207  ( HP 7900 DISC I/O                                    HN/780203   8 OF 11 )
1208  ( HP 7900 DISC I/O                                    JV/780330   9 OF 11 )
1209  ( HP 7900 DISC I/O                                    JV/780330  10 OF 11 )
1210  ( HP 7900 DISC I/O                                    HN/780320  11 OF 11 )
1290  ( STERREWACHT UTRECHT CONFIGURATION                   JV/780403   1 OF  1 )
1300  ( LINE PRINTER DRIVER                                 HN/780321   1 OF ** )
1301  ( LINE PRINTER DRIVER : NON-INTERRUPT                 JV/780330   2 OF ** )
1302  ( LINE PRINTER DRIVER : INTERRUPT                     JV/780330   3 OF ** )
1303  ( LINE PRINTER DRIVER                                 JV/780403   4 OF ** )
1310  ( VDU DRIVER                                          HN/780320   1 OF ** )
1311  ( VDU DRIVER  NON-INTERRUPT                           JV/780330   2 OF ** )
1312  ( VDU DRIVER  INTERRUPT                               HN/780320   3 OF ** )
1313  ( VDU DRIVER  INTERRUPT                               JV/780330   4 OF ** )
1314  ( VDU DRIVER                                          JV/780403   5 OF ** )
1400  ( 32-BIT OPERATORS                                    HN/780315   1 OF  4 )
1401  ( 32-BIT OPERATORS                                    HN/780315   2 OF  4 )
1402  ( 32-BIT OPERATORS                                    HN/780315   3 OF  4 )
1403  ( 32-BIT OPERATORS                                    HN/780321   4 OF  4 )
1500  ( MISCELLANEOUS                                       JV/780403   1 OF ** )
1501  ( MISCELLANEOUS                                       HN/780315   2 OF ** )
1502  ( MISCELLANEOUS                                       JV/780320   3 OF ** )
1503  ( MISCELLANEOUS                                       HN/780315   4 OF ** )
1600  ( XY STORAGE DISPLAY                                  JV/771010   1 OF 12 )
1601  ( XY STORAGE DISPLAY                                  HN/780315   2 OF 12 )
1602  ( XY STORAGE DISPLAY                                  JV/780330   3 OF 12 )
1603  ( XY STORAGE DISPLAY                                  JV/771010   4 OF 12 )
1604  ( XY STORAGE DISPLAY                                  JV/771010   5 OF 12 )
1605  ( XY STORAGE DISPLAY                                  JV/771010   6 OF 12 )
1606  ( XY STORAGE DISPLAY                                  JV/771010   7 OF 12 )
1607  ( XY STORAGE DISPLAY                                  JV/771010   8 OF 12 )
1608  ( XY STORAGE DISPLAY                                  JV/771010   9 OF 12 )
1609  ( XY STORAGE DISPLAY                                  JV/771010  10 OF 12 )
1610  ( XY STORAGE DISPLAY                                  JV/771010  11 OF 12 )
1611  ( XY STORAGE DISPLAY                                  JV/771010  12 OF 12 )
1620  ( UTILITY SAVE/RESTORE                                JV/780403   1 OF ** )
1630  ( UTILITY GENERATE STERREWACHT SYSTEM                 JV/780403   1 OF ** )
```

```
BLOCK 13(15B)
 0. ( BASIC FORTH ASSEMBLER FOR HP 2100        JV/780313   1 OF ** )
 1.    0 CONSTANT 0      1 CONSTANT 1     2 CONSTANT 2
 2.
 3. ASSEMBLER DEFINITIONS
 4. ' NEXT,      @  3 + CPU .NEXT,
 5. ' NEXT,      @ -1 + CPU PUT,
 6. ' PUT,       @ -4 + CPU PUSH,
 7. ' 2POP,      @  2 + CPU POP,
 8. ' POP,       @ 13 + CPU BINARY,
 9. 101 CONSTANT SP     SP 1 + CONSTANT SP1
10. SP ) CONSTANT S)    SP1 ) CONSTANT S1)
11. 0 CONSTANT A   1 CONSTANT B   2 CONSTANT U   3 CONSTANT OWNER
12.
13. 104 CONSTANT N        ( SEMI-TEMPORARY STORAGE AREA )
14. N 3 + CONSTANT BASE0
15. 1 +BLOCK CONTINUED

BLOCK 14(16B)
 0. ( BASIC FORTH ASSEMBLER FOR HP 2100        JV/780330   2 OF ** )
 1. 10 MR AND,  14 MR JSB,  20 MR XOR,  24 MR JMP,  30 MR IOR,
 2. 34 MR ISZ,  40 MR ADA,  44 MR ADB,  50 MR ANE,  54 MR BNE,
 3. 60 MR LDA,  64 MR LDB,  70 MR STA,  74 MR STB,
 4. 2002 CPU A0,   6002 CPU B0,   2040 CPU E0,   2001 CPU RSS,
 5. 2020 CPU A+,   6020 CPU B+,   1727 CPU AL8,  1222 CPU 2RAL,
 6. 5727 CPU BL8,  5002 CPU BLS,  6004 CPU INB,  4022 CPU RBL,
 7. 5121 CPU 2BRS, 5222 CPU 2RBL, 2004 CPU INA,  1000 CPU ALS,
 8.   65 CPU CAER, 4065 CPU CBER, 3004 CPU TCA,  7004 CPU TCB,
 9. 2100 CPU CLE,  1600 CPU ELA,  2400 CPU CLA,  6400 CPU CLB,
10. 3000 CPU CMA,  7000 CPU CMB,  2021 CPU A-,   6021 CPU B-,
11. 1121 CPU 2ARS,
12. 27 I/O STC,  67 I/O CLC,  20 I/O HLT,  25 I/O LIA,  65 I/O LIB,
13. 26 I/O OTA,  66 I/O OTB,  24 I/O MIA,  21 I/O STF,  31 I/O CLF,
14. 23 I/O SFS,  22 I/O SFC,  64 I/O MIB,
15. 1 +BLOCK CONTINUED

BLOCK 15(17B)
 0. ( BASIC FORTH DEFINITIONS                  HN/780310   3 OF ** )
 1. FORTH DEFINITIONS
 2. CODE -       S) LDA,  TCA,      S1) ADA,  BINARY,      EDOC
 3. CODE MINUS   S) LDA,  TCA,                PUT,         EDOC
 4. CODE AND     S) LDA,  S1) AND,            BINARY,      EDOC
 5. CODE XOR     S) LDA,  S1) XOR,            BINARY,      EDOC
 6. CODE OR      S) LDA,  S1) IOR,            BINARY,      EDOC
 7. CODE 1-      S) LDA,  -1 # ADA,           PUT,         EDOC
 8. CODE SP@     SP LDA,                      PUSH,        EDOC
 9. CODE 2/      S) LDA,  21 ( ARS ) ,        PUT,         EDOC
10. CODE 2*      S) LDA,  20 ( ALS ) ,        PUT,         EDOC
11. CODE COM     S) LDA,  CMA,                PUT,         EDOC
12. CODE PICK    S) LDA,  SP ADA,  A ) LDA,   PUT,         EDOC
13. CODE +!      S1) LDA, S) LDB,  B ) ADA,   B ) STA,  2POP, EDOC
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 16(20B)
 0. ( BASIC FORTH DEFINITIONS                          HN/780208   4 OF ** )
 1.    : SET    SWAP CONSTANT , ;CODE
 2.                                    INB,  B ) LDA,  INB,  B ) LDB,
 3.                                    B ) STA,  NEXT,              EDOC
 4.    : BYTE   DUP + ;
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 17(21B)
 0. ( BASIC FORTH DEFINITIONS                          HN/780310   5 OF ** )
 1. : USER  CONSTANT ;CODE  INB,  B ) LDA,  U ADA,  PUSH,    EDOC
 2. : UABS  SYSTEM + CONSTANT ;
 3. : UREL  CONSTANT ;
 4.      0 UABS STATUS        1 UABS FOLLOWER      2 UABS PC
 5.      3 USER TIME-OUT   ( 4 UABS SP )           5 UABS RP
 6.      6 UABS SP0           7 UABS RP0           10 UABS DP
 7.     11 UABS STATE        12 UABS CURRENT       13 UABS CONTEXT
 8.     14 UABS LAST         15 UABS OP         ( 16 UABS IP )
 9.     17 UABS IN           20 UABS SCR           21 UABS MSG
10.     22 UABS DPL          23 UABS BASE          24 USER ACCEPT-IC
11.     25 USER TYPE-IC      26 USER VOC-LINK      27 USER FLD
12.     30 USER WRITE-IC     31 USER BLK           32 USER ?F
13.     33 USER DUMP-OFFSET  34 USER WARNING       35 USER CSP
14.     36 USER BLK-LOW      37 USER BLK-HIGH      40 USER PV-ADDRESS
15. 1 +BLOCK CONTINUED

BLOCK 18(22B)
 0. ( BASIC FORTH STATE AND PRECEDENCE                 HN/780208   6 OF ** )
 1. CODE @STATE  STATE LDA,  PUSH,                    EDOC
 2. CODE !STATE  S) LDA,  STATE STA,  POP,            EDOC
 3.
 4. ' ' 3 - DUP @ 77777 AND SWAP !   ( CLEAR PRECEDENCE OF ' )
 5.
 6. : 1IMP   ' 3 -  DUP @  100000 OR   SWAP ! ;
 7.
 8.      0  STATE  SET  [              ( EXECUTE )
 9. 100000  STATE  SET  ]              ( COMPILE )
10.
11. : 'C,    @STATE [ ' 1-    , !STATE ;  ( COMPILE NEXT WORD )
12. : 'C     @STATE ' ' 1- SWAP !STATE ;
13. 1IMP '   1IMP [   1IMP 'C,   1IMP 'C
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 19(23B)
 0. ( BASIC FORTH                                    JV/780120   7 OF ** )
 1. 10  BASE  SET  OCTAL              12  BASE  SET  DECIMAL
 2. 20  BASE  SET  HEX      ( HEX IS VALID ONLY FOR OUTPUT )
 3. CODE HERE  DP LDA,  PUSH,                        EDOC
 4. CODE HEAD  CURRENT LDA,  PUSH,                   EDOC
 5. CODE ?S    SP LDA,  TCA,  SP0 ADA,  PUSH,        EDOC
 6. CODE ROT   SP1 LDB,  INB,  B ) LDA,  N STA,  S1) LDA,
 7.            B ) STA,  N LDA,  S) LDB,  S1) STB,  PUT,  EDOC
 8. CODE DP+1  S) LDA,  DP ADA,  DP STA,  POP,  EDOC
 9.
10. : ORPHAN   'C,  ASSEMBLER  HERE  HERE 1+ ,  ;
11. : ADOPT    ,  ;                                 IIMP ADOPT
12. ' CODE 1- @
13. : :ORPHAN  HEAD @ LAST ! HERE LIT ADOPT , ] ;
14. : ,CODE    CODE HERE 1- +! ;
15. 1 +BLOCK CONTINUED

BLOCK 20(24B)
 0. ( BASIC FORTH                                    JV/780313   8 OF ** )
 1. CODE \      IN LDA,  -1 # ADA,  177700 # AND,
 2.             100 # ADA,  IN STA,  NEXT,  EDOC
 3. IIMP \      \ INTERPRETS REMAINDER OF LINE AS COMMENT !!!!!
 4.
 5. ORPHAN ( LOC )   N 1+ LDA,  PUSH,        EDOC
 6. : ?DEF             40 WORD 0 FIND DROP ADOPT ;
 7.
 8. CODE R>    RP ) LDB,  RP ISZ,  RP ) LDA,
 9.            RP ) STB,  PUSH,    EDOC
10.
11. CODE >R    RP ) LDB,  S) LDA,  RP ) STA,
12.            RP LDA,  -1 # ADA,  RP STA,
13.            RP ) STB,  POP,     EDOC
14.
15. 1 +BLOCK CONTINUED

BLOCK 21(25B)
 0. ( BASIC FORTH ASSEMBLER MACRO'S                  JV/780330   9 OF ** )
 1. ASSEMBLER DEFINITIONS
 2. : NOT     HERE 1- DUP @ 1 XOR SWAP ! ;
 3. : CLF     HERE 1- DUP @ 1000 OR SWAP ! ;
 4. : SUB,    TCA,  ADA,  TCA,  ;
 5. : ISP,    SP ISZ,  SP1 ISZ,  ;
 6. : DSP,    SP LDB,  SP1 STB,  -1 # ADB,  SP STB,  ;
 7. : ADM,    DUP ADA,  STA,  ;
 8. : IF,     HERE  0 ,  ;
 9. : THEN,   HERE SWAP DP !  DUP JMP,  DP !  ;
10. : ELSE,   IF,  SWAP  THEN,  ;
11. FORTH DEFINITIONS
12. : SUBROUTINE  0 VARIABLE  'C,  ASSEMBLER ;
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 22(26B)
 0. ( BASIC FORTH COMPILER DIRECTIVES          JV/780330  10 OF ** )
 1. ORPHAN ( SKIP )   RP ) LDA, A ) LDA, RP ) STA, NEXT,     EDOC
 2. ORPHAN ( SKIP? )  S) LDA, A0, IF, RP ) LDA, A ) LDA, RP ) STA,
 3.                                  POP,
 4.                                THEN, RP ) ISZ, POP,      EDOC
 5.       : THEN   HERE SWAP ! ;
 6. OVER : ELSE   LIT ADOPT  , HERE 0 , SWAP THEN ;
 7. OVER : REPEAT LIT ADOPT  , HERE 1+ SWAP ! , ;
 8. DUP  : IF     LIT ADOPT  , HERE 0 , ;
 9.       : BEGIN  HERE ;
10. DUP  : END    LIT ADOPT  , , ;
11. 2DROP
12. : WHILE  IF  ;
13. 1IMP THEN    1IMP ELSE    1IMP REPEAT  1IMP IF
14. 1IMP BEGIN   1IMP END     1IMP WHILE
15. 1 +BLOCK CONTINUED

BLOCK 23(27B)
 0. ( BASIC FORTH COMPILER DIRECTIVES          JV/780119  11 OF ** )
 1. ORPHAN ( DO )   RP ) LDB, S1) LDA, RP ) STA, -2 # LDA,
 2.                 RP ADM,  A ) STB,  INA,  S) LDB,  A ) STB,
 3.                 2POP,                                EDOC
 4. ORPHAN ( LOOP ) RP LDA,  INA,  A ) ISZ,  0 ,
 5.                 A ) LDB,  INA,  CMB,  A ) ADB,  B+,
 6.                 IF,  RP ) LDA,  A ) LDA,  RP ) STA,  NEXT,
 7.                 THEN,
 8.                 RP ) LDB,  INB,  RP STA,  RP ) STB,  NEXT,
 9.                 EDOC
10. ORPHAN ( +LOOP ) RP LDB,  INB,  S) LDA,  B ) ADM,  CMA,  INB,
11.                 B ) ADA,  S) XOR,  A+,
12.                 IF,  RP ) LDB,  B ) LDB,  RP ) STB,  POP,
13.                 THEN,  RP ) LDA, INA, RP STB, RP ) STA, POP,
14.                 EDOC
15. 1 +BLOCK CONTINUED

BLOCK 24(30B)
 0. ( BASIC FORTH COMPILER DIRECTIVES          HN/780310  12 OF ** )
 1.
 2. ; +LOOP    LIT ADOPT  , , ;                 1IMP +LOOP
 3. : LOOP     LIT ADOPT  , , ;                 1IMP LOOP
 4. : DO       LIT ADOPT  , HERE ;              1IMP DO
 5.
 6. CODE I      RP LDB,  1 # ADB,  B ) LDA,  PUSH,    EDOC
 7. CODE J      RP LDB,  3 # ADB,  B ) LDA,  PUSH,    EDOC
 8. CODE K      RP LDB,  5 # ADB,  B ) LDA,  PUSH,    EDOC
 9. CODE LEAVE RP LDA,  1 # ADA,  A ) LDB,
10.                     1 # ADA,  A ) STB,  NEXT,    EDOC
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 25(31B)
 0. ( BASIC FORTH COMPARE OPERATORS              JV/780120  13 OF ** )
 1. CODE 0<    S) LDA,  0 ) # AND,  22 ( RAL ) ,  PUT,     EDOC
 2. CODE 0=    S) LDA,  A0,  3400 ( CCA ) ,  TNA,  PUT,    EDOC
 3. CODE <     S) LDA,  1 CLF,  TCA,  S1) ADA,  1 SFC,
 4.            CMA,  22 ( RAL ) ,  1 # AND,  BINARY,       EDOC
 5.
 6. : >     SWAP < ;
 7. : =     - 0= ;
 8. : 0>    MINUS 0< ;
 9.
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 26(32B)
 0. ( BASIC FORTH CONDITIONAL INTERPRETATION  JV/771201  14 OF ** )
 1. CONTEXT @ @                          \ HEAD OF DICTIONARY
 2. CODE IFEND          EDOC             \ FIRST ENTRY IN COND. DICT.
 3. HERE 2 -                             \ ADDRESS OF LINK FIELD
 4. CODE OTHERWISE      EDOC             \ LAST  ENTRY IN COND. DICT.
 5. HERE CONTEXT @ ! ,                   \ HEAD OF COND. DICT.
 6. ROT CONTEXT @ !                      \ SKIP COND. DICT.
 7. 0 ROT !                              \ LIMIT COND. DICT.
 8. : OTHERWISE  CONTEXT @               \ SAVE HEAD OF DICTIONARY
 9.              LIT ADOPT CONTEXT !     \ SWITCH TO COND. DICT.
10.              BEGIN ?DEF END          \ READ UNTIL KNOWN WORD
11.              CONTEXT !               \ RESTORE HEAD OF DICTIONARY
12.              ;
13. : IFTRUE     0= IF OTHERWISE THEN ;
14. : IFEND      ;
15. 1 +BLOCK CONTINUED

BLOCK 27(33B)
 0. ( BASIC FORTH FIX COMPUTER TYPE              HN/780310  15 OF ** )
 1. ' SYSTEM                                  \ ADDRESS OF SYSTEM AREA
 2. 1+ DUP  @ CONSTANT #CHF                   \ CHARACTER OFFSET
 3. 1+ DUP  @ CONSTANT COMPUTER               \ COMPUTER TYPE
 4.         ASSEMBLER DEFINITIONS             \ FOR DEF. OF MPY, DIV.
 5.         COMPUTER 2 AND                    \ HARDWARE OR SOFTWARE ?
 6.         IFTRUE  2 DEAU MPY,               \ HARDWARE MPY,
 7.                 4 DEAU DIV,               \ HARDWARE DIV,
 8.         OTHERWISE                         \ SOFTWARE ROUTINES
 9. 1+ DUP @ DUP ) : MPY, LIT ADOPT JSB, , ; \ SOFTWARE MPY,
10.          1+  ) : DIV, LIT ADOPT JSB, , ; \ SOFTWARE DIV,
11.         IFEND
12.         FORTH DEFINITIONS
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 28(34B)
 0. ( BASIC FORTH FIX COMPUTER TYPE                JV/780201  16 OF ** )
 1. 1+ DUP  @ CONSTANT MASS-STORAGE          \ MASS-STORAGE TYPE
 2.           MASS-STORAGE 1 AND ?F !        \ FAST/SLOW MSDEV
 3. 1+ DUP  @ CONSTANT U1#                   \ SAVE/RESTORE LENGTH
 4. 1+ DUP  @ CONSTANT U2#                   \ INDEXED LENGTH
 5. DROP                                     \ VALUE ON STACK
 6. 1 +BLOCK CONTINUED
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 29(35B)
 0. ( BASIC FORTH ARITHMETIC                       JV/771201  17 OF ** )
 1. CODE ABS      S) LDA,  A+,  TCA,  PUT,                          EDOC
 2. CODE */       SP1 LDA, INA, A ) LDA, S1) MPY,
 3.               S) DIV,  ' BINARY, @ 2 - , ( 2BINARY, )           EDOC
 4. CODE *        S) LDA,  S1) MPY,  BINARY,                        EDOC
 5. CODE /MOD     S1) LDA, CLB, A+, CMB,
 6.               S) DIV,  S1) STB,  PUT,                           EDOC
 7. :    /        /MOD  SWAP  DROP  ;
 8. :    MOD      /MOD  DROP  ;
 9. CODE MAX      S1) LDA, TCA, 1 CLF, S) ADA,
10.           HERE  1 SFC,  CMA,  A+,  POP,  S) LDA,
11.               BINARY,                                           EDOC
12. CODE MIN      S) LDA,  TCA,  1 CLF,  S1) ADA,
13.           JMP,                                                  EDOC
14. 1 +BLOCK CONTINUED
15.

BLOCK 30(36B)
 0. ( BASIC FORTH VOCABULARY HANDLING          HN/780310  18 OF ** )
 1. : VOCABULARY  HERE  0 CONSTANT -2 DP+!   \ VOCABULARY ENTRY
 2.               100000 HERE 2 - +!         \ SET PRECEDENCE BIT
 3.               LIT [ ' FORTH 1- @ , ] ,   \ SET CODE ADDRESS
 4.               1600 DUP , HERE SWAP ,     \ PSEUDO ENTRY
 5.               [ #CHF 3 > IFTRUE
 6.                 #CHF 3 - 2/ ]
 7.               LIT ADOPT DP+! [ IFEND ]   \ FINISH ENTRY
 8.               , ,                        \ VOC. HEAD AND LINK
 9.               HERE 1- VOC-LINK @ ,       \ SET CHAIN
10.               VOC-LINK !                 \ UPDATE HEAD OF CHAIN
11.               ;
12. 1IMP VOCABULARY
13. : CHAIN       'C, ' HEAD 1+ ! ;          \ CHAIN A VOCABULARY
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 31(37B).
 0. ( BASIC FORTH VOCABULARY HANDLING          JV/780330  19 OF ** )
 1. : FORGET ?DEF DUP 0= IF 17 ERROR THEN
 2.          VOC-LINK @ >R
 3.          BEGIN DUP I <
 4.          WHILE 'C, FORTH DEFINITIONS
 5.                R> 1+ @ >R I VOC-LINK !
 6.          REPEAT
 7.          BEGIN I @ OVER >                  \ CHAIN TOO HIGH?
 8.                IF ' FORTH I ! THEN         \ CHAIN TO FORTH
 9.                I LIT [ #CHF 3 MAX 3 + 2/ , ] -
10.                BEGIN [ #CHF IFTRUE #CHF 1+ 2/ ] LIT ADOPT
11.                      [ OTHERWISE ] DUP COUNT SWAP DROP
12.                        3 MAX 2/ 1+
13.                      [ IFEND ]
14. [ 1 +BLOCK CONTINUED
15.

BLOCK 32(40B)
 0. ( BASIC FORTH VOCABULARY HANDLING          JV/780118  20 OF ** )
 1.  ]
 2.                    + @ OVER OVER >
 3.                END
 4.                I 1- !
 5.                R> 1+ @ DUP >R 0=
 6.          END
 7.          R> DROP DP ! ;
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 33(41B)
 0. ( BASIC FORTH                              JV/780313  21 OF ** )
 1. CODE <#      S) LDA,  BASE0 STA,
 2.              SP LDA, -3 # ADA, 22 ( RAL ) , -1 # ADA, DP STA,
 3.              CLA,  N STA,
 4.              POP,                                    EDOC
 5. SUBROUTINE -DEPOSIT DP LDA,  CAER,
 6.                     E0, NOT  BL8,
 7.                     E0, NOT  A ) ADB,
 8.                     A ) STB,
 9.                     ELA,  -1 # ADA,  DP STA,
10.                     N ISZ,
11.                     -DEPOSIT ) JMP,             EDOC
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 34(42B)
 0. ( BASIC FORTH                                    JV/780119  22 OF ** )
 1. CODE DIGIT   DPL LDA,  N ANE,  RSS,
 2.              IF,  56 # LDB,  -DEPOSIT JSB,
 3.              THEN,
 4.              S) LDA,  CLB,
 5.              BASE0 DIV,  S) STA,
 6.              S1) LDA,  BASE0 DIV,  S1) STA,
 7.              -12 # ADB,  B-,  7 # ADB,
 8.              72 # ADB,
 9.              -DEPOSIT JSB,
10.              NEXT,                          EDOC
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 35(43B)
 0. ( BASIC FORTH                                    HN/780119  23 OF ** )
 1. CODE SIGN     ISP,  S1) LDA,  55 # LDB,
 2.               A+,  -DEPOSIT JSB,
 3.               NEXT,                                          EDOC
 4. CODE #>       OP LDA,  INA,  S1) STA,  N LDA,  PUT,  EDOC
 5. : DIGITS      BEGIN DIGIT OVER ABS OVER + 0= END
 6.               SIGN ;
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 36(44B)
 0. ( BASIC FORTH                                    HN/780212  24 OF ** )
 1. CODE LFILL    ' FLD LDA,  U ADA,  A ) LDA,  N LDB,
 2.               TCA,  B ADA,
 3.               A-, NEXT,
 4.               N 1+ STA,
 5.               HERE   40 # LDB,
 6.                      -DEPOSIT JSB,
 7.                      N 1+ ISZ,
 8.               JMP,
 9.               NEXT,
10.               EDOC
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.
```

```
BLOCK 37(45B)
 0. ( BASIC FORTH                                    PN/780212  25 OF ** )
 1. CODE WRITE  ' WRITE-IC LDB,  U ADB,  B ) LDB,
 2.             S) LDA,  A0,  .NEXT,  2POP,  EDOC
 3. : TERMINAL  TYPE-IC @ WRITE-IC ! ;  TERMINAL
 4. : FREE    0 FLD !  -1 DPL ! ;
 5. : WCH     SP@  COUNT 1+ WRITE DROP ;
 6. : CR      15 WCH  12 WCH ;
 7. : SPACE   40 WCH ;
 8. : .SD     <# DIGITS LFILL #> WRITE ;
 9. : S.      DUP ABS 0 BASE @ .SD ;
10. : .       FREE S. SPACE ;
11. : ?       @ . ;
12. : D.      0 SWAP 0 10 .SD ;
13. : T.      DUP ABS 0 12 .SD ;
14. : SPACES 0 DO SPACE LOOP ;
15. 1 +BLOCK CONTINUED

BLOCK 38(46B)
 0. ( BASIC FORTH                                    JV/780330  26 OF ** )
 1. : LINE        17 MIN 0 MAX 40 *
 2.               BLK @  BLOCK + ;
 3. : -TRAILING 0 SWAP 0
 4.               DO OVER I + B@ DUP
 5.                  IF 40 -
 6.                     IF DROP I THEN
 7.                  ELSE DROP
 8.                  THEN
 9.               LOOP  1+  ;
10. : LINEWRITE LINE BYTE 100
11.               -TRAILING
12.               WRITE
13.               TEG              \ RELEASE BLOCK BUFFERS
14.               ;
15. 1 +BLOCK CONTINUED

BLOCK 39(47B)
 0. ( BASIC FORTH                                    JV/780119  27 OF ** )
 1. ORPHAN      S) LDB,  RP LDA, INA,  A ) ADB,
 2.             A ) STB,  POP,                              EDOC
 3. :ORPHAN ( E." ) I COUNT DUP 2/ 1+ [ SWAP ] ADOPT
 4.                 BEGIN WRITE ;
 5. : ."        @STATE IF LIT [ ROT ] ADOPT , THEN
 6.             42 WORD
 7.             HERE COUNT  @STATE  END
 8.             2/ 1+ DP+! DROP ;
 9. !IMP ."
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 40(50B)
 0. ( BASIC FORTH                                    HN/780310  28 OF ** )
 1. : LIST      FREE  DUP  BLK !  CR
 2.             ." BLOCK " DUP T. ." (" 0. ." B)"
 3.             3 FLD !  0 DPL !  20 @
 4.             DO CR  I S.  SPACE
 5.                    I LINEWRITE
 6.             LOOP
 7.             CR FREE ;
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 41(51B)
 0. ( BASIC FORTH                                    JV/780330  29 OF ** )
 1. #CHF IFTRUE
 2.   : ID.     SPACE COUNT SWAP OVER
 3.             #CHF MIN WRITE
 4.             BEGIN DUP #CHF >
 5.             WHILE ." ↑" 1-
 6.             REPEAT
 7.             DROP ;
 8.       OTHERWISE
 9.   : ID.     SPACE COUNT WRITE ;
10.       IFEND
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 42(52B)
 0. ( BASIC FORTH                                    HN/780310  30 OF ** )
 1. : DLIST     BEGIN CR 5 FLD !
 2.                   DUP S. DUP COUNT S. DROP DUP ID.
 3.                   [ #CHF IFTRUE #CHF 1+ 2/ ] LIT ADOPT
 4.                   [ OTHERWISE ] DUP COUNT 2/ 1+ 2 MAX SWAP DROP
 5.                   [ IFEND ]
 6.                   + @  DUP 0=
 7.             END
 8.             DROP ;
 9. : VLIST     CONTEXT @ @ DLIST ;
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 43(53B)
 0. ( BASIC FORTH ERROR HANDLING                    JV/780316  31 OF ** )
 1. : MESSAGE  1- 17 /MOD 132 + ABLOCK
 2.            SWAP 1+ 40 * + BYTE 100 -TRAILING
 3.            WRITE TEG ;
 4. HERE  )  WRITE-IC @ SWAP  TERMINAL
 5.          ?F @ DUP 0< IF ABORT THEN
 6.              IF MESSAGE
 7.              ELSE ." ERROR# " T.
 8.              THEN
 9.          SPACE
10.          WRITE-IC !  ABORT ;
11. ASSEMBLER HERE ! ERROR, @  1777 AND ! SCR LDB,  SP0 ) STB,
12.           SP0 LDB, SP STB,  RP0 LDB, -1 # ADB,  RP STB,
13.           # LDB,  RP ) STB,  PUSH,    EDOC
14. 1 +BLOCK CONTINUED
15.

BLOCK 44(54B)
 0. ( BASIC FORTH                                   JV/780125  32 OF ** )
 1. CODE MOVE   S) LDA,  TCA,  ISP,
 2.             S1) LDB, N STB,
 3.             S) LDB, N 1+ STB,
 4.             HERE  N ) LDB,  N ISZ,
 5.                   N 1+ ) STB, N 1+ ISZ,
 6.                   A ISZ,
 7.             JMP,
 8.             2POP,
 9.             EDOC
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 45(55B)
 0. ( BASIC FORTH                                   HN/780212  33 OF ** )
 1. ORPHAN ( MS )   S) LDA, TCA,
 2.                 HERE  -307 # LDB,
 3.                       HERE  B ISZ,  JMP,
 4.                       A ISZ,
 5.                 JMP,
 6.                 POP,
 7.                 EDOC
 8. : MS    ADOPT ;
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 46(56B)
 0. ( BASIC FORTH                                    JV/780316  34 OF ** )
 1. :  ;;  CONSTANT  R> ,  ;CODE
 2.                RP LDA,  -1 # ADA,  RP STA,
 3.                INB,  B ) LDA,
 4.                INB,  B ) LDB,  RP ) STB,
 5.                PUSH,
 6.                EDOC
 7. : LOADER   ;; LOAD ;
 8. : DUMP      7 FLD ! OVER + SWAP
 9.                DO CR I S. I 10 + I
10.                   DO I DUMP-OFFSET @ + @ 0.
11.                   LOOP
12.             10 +LOOP
13.             0 DUMP-OFFSET ! ;
14. CODE PAUSE   PAUSE,  NEXT,     EDOC
15. 1 +BLOCK CONTINUED

BLOCK 47(57B)
 0. ( BASIC FORTH                                    JV/780403  35 OF ** )
 1. DECIMAL
 2. 80 LOADER RESIDENT
 3. RESIDENT SECURITY
 4. OCTAL  : PAGE 14 WCH ;  DECIMAL     MISCELLANEOUS
 5. STERREWACHT
 6. CLOCK
 7. 1EDITOR
 8. CR ." F77 UTRECHT  30 MAR 1978"
 9. CR ." MEMORY LEFT " SP@ HERE - DUP 0. ." B  " T. ." WORDS"
10. CR
11. ;S
12.
13.
14.
15.

BLOCK 80(120B)
 0. ( RESIDENT BLOCKS                                JV/780403   1 OF ** )
 1. FORTH DEFINITIONS  BASE @ DECIMAL
 2. : RESIDENT ;
 3.  100 LOADER BLOCK-ASSEMBLER
 4.  110 LOADER 21MX-ASSEMBLER
 5.  300 LOADER METAFORTH
 6. 1000 LOADER SECURITY
 7. 1020 LOADER MONITOR
 8. 1050 LOADER CLOCK
 9. 1100 LOADER PV-DEFINE
10. 1110 LOADER FILES
11. 1120 LOADER EDITOR
12. 1130 LOADER 1EDITOR
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 81(121B)
 0. ( RESIDENT BLOCKS                                  JV/780403   2 OF ** )
 1. 1200 LOADER DISC-DRIVER
 2. 1290 LOADER STERREWACHT
 3. 1300 LOADER LP-DRIVER
 4. 1310 LOADER VDU-DRIVER
 5. 1500 LOADER MISCELLANEOUS
 6. BASE ! ;S
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 90(132B)
 0. ( FORTH [ ERROR ] MESSAGES   1 - 15          JV/780120   1 OF  2 )
 1.  DICTIONARY OVERFLOW
 2.  STACK UNDERFLOW
 3.  BASE PAGE TABLE OVERFLOW
 4.
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.  ILLEGAL WORD OR NUMBER
15.  WORD NOT FOUND

BLOCK 91(133B)
 0. ( FORTH [ ERROR ] MESSAGES  16 - 30          JV/780403   2 OF  2 )
 1.  COMPILATION ONLY
 2.  EXECUTION ONLY
 3.  NOT PAIRED
 4.  DEFINITION NOT FINISHED
 5.  NO REDEFINITION OF 'EOT' ALLOWED
 6.  INSUFFICIENT CORE
 7.  DEFINITIONS NOT ALLOWED
 8.  NOT USED DURING BLOCK LOADING
 9.  BLOCK NOT IN FILE
10.  PSEUDO VECTORS NOT YET AVAILABLE, TYPE PV-DEFINE
11.  NOT ENOUGH WORK SPACE ON TOP OF DICTIONARY
12.
13.
14.
15.
```

```
BLOCK 97(141B)
 0. ( TEMPORARY CONVERSION TO STANDARD FORTH  JV/780321   1 OF  3 )
 1. BASE @ OCTAL
 2. : +BLOCK *BLOCK ; : *BLOCK ABORT ; IMP *BLOCK
 3. ASSEMBLER DEFINITIONS : DP DP ; FORTH DEFINITIONS
 4. : DP+! DP +! ;   : DP ABORT ; IMP DP
 5. : BUFFER BLOCK UPDATE ;
 6. : 2* 2 * ;
 7. : WORD   DELIMITER ! WORD ;
 8. : 'C, @STATE 0 !STATE [ IMP ' ] ' [ IMP ' ] SWAP !STATE 1= , ;
 9. IMP 'C,  : COMPILE ABORT , ; IMP COMPILE
10. : VARIABLE INTEGER ; : INTEGER ABORT ; IMP INTEGER
11. : SPACE BLANK ; : BLANK ABORT ; IMP BLANK
12. : ) I) ; : I) ABORT ; IMP I)
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 98(142B)
 0. ( TEMPORARY CONVERSION TO STANDARD FORTH  JV/780330   2 OF  3 )
 1. 0 VARIABLE LLLL : LAST LAST @ 2 - 1- LLLL ! LLLL ;
 2. 3 CONSTANT #CHF   : 0 ABORT ;  IMP 0
 3. : 2/ 2 / ;
 4. : ." 'C, " ; IMP ." : " ABORT ; IMP "
 5. ASSEMBLER DEFINITIONS 20 I/O HLT, FORTH DEFINITIONS
 6. : FDOC ;
 7. : DPL 0 ; : 0 ABORT ; IMP 0
 8. : DUMP OVER + DUMP ;
 9. : LEAVE 'C, EXIT ; IMP LEAVE : EXIT ABORT ; IMP EXIT
10.  1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 99(143B)
 0. ( TEMPORARY CONVERSION TO STANDARD FORTH  JV/780315   3 OF  3 )
 1. : NDROP 0 DO DROP LOOP ;
 2. CODE PICK S) LDA, SP ADA, A ) LDA, PUT ,
 3. USER CONSTANT U
 4. ASSEMBLER DEFINITIONS
 5. NEXT CPU NEXT,    POP. CPU 2POP,  POP CPU POP,
 6. PUSH CPU PUSH,    PUT  CPU PUT,
 7. USER CONSTANT U
 8. FORTH DEFINITIONS
 9. 21 CONSTANT XYS
10. BASE ! ;S
11.
12.
13.
14.
15.
```

```
BLOCK 100(144B)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS              JV/770926   1 OF  6 )
 1. FORTH DEFINITIONS
 2. VOCABULARY BLOCK-ASSEMBLER      BLOCK-ASSEMBLER DEFINITIONS
 3. BASE @ OCTAL
 4. 0 VARIABLE BK                                 \ ASSEMBLY IN BLOCK BK
 5. 0 VARIABLE BLOCK-ADDRESS                      \ ADDRESS  IN BLOCK BK
 6. 0 VARIABLE ORIGIN                             \ CODE ORIGIN
 7. : CHECK-ADDRESS                               \ SHOULD NOT EXCEED BLOCK
 8.                BLOCK-ADDRESS @
 9.                DUP  777 >  SWAP 0<   OR
10.                IF ." BLOCK OVERFLOW" ABORT THEN
11.                ;
12.
13. 0 CONSTANT A                                  \ A REGISTER
14. 1 CONSTANT B                                  \ B REGISTER
15. 1 +BLOCK CONTINUED

BLOCK 101(145B)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS              JV/770926   2 OF  6 )
 1. : ,              \ COMPILE VALUE ON STACK INTO BLOCK BK
 2.            BK @ BLOCK BLOCK-ADDRESS @ + !          \ COMPILE
 3.            1 BLOCK-ADDRESS +!                      \ INCREMENT
 4.            CHECK-ADDRESS                           \ CHECK ADDRESS
 5.            UPDATE                                  \ BLOCK CHANGED
 6.            ;
 7.
 8. : MR             \ ASSEMBLE MEMORY REFERENCE INSTRUCTION
 9.            1000 * ;:  SWAP                         \ INSTR. CODE
10.            DUP 101777 AND SWAP 76000 AND           \ ADDRESS MASK
11.            IF  2000 OR  THEN                       \ CURRENT PAGE
12.            OR  ,                                   \ COMPILE INSTR.
13.            ;
14. 1 +BLOCK CONTINUED
15.

BLOCK 102(146B)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS              JV/770926   3 OF  6 )
 1. : CPU            \ COMPILE CPU INSTRUCTION
 2.            ;: ,                                    \ COMPILE INSTR.
 3.            ;
 4.
 5. : )        100000 OR ;
 6.
 7. : I/O            \ COMPILE I/O INSTRUCTION
 8.            100 * )  ;:                             \ INSTR. CODE
 9.            OR ,                                    \ ADD SC, COMPILE
10.            ;
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 103(1478)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS                JV/770926   4 OF  6 )
 1. : !,             \ STORE IN A BLOCK
 2.          ORIGIN @ -                             \ ADDRESS IN BLOCK
 3.          DUP 777 > OVER 0< OR                   \ ADRRESS OK?
 4.          IF ." ADDRESS ERROR" ABORT THEN
 5.          BK @ BLOCK + !                         \ CHANGE BLOCK
 6.          UPDATE ;                               \ CONTENTS CHANGED.
 7.
 8. : HERE  ORIGIN @  BLOCK-ADDRESS @ + ;  \ PUSH CORE ADDRESS
 9.
10. : VAR           \ DEFINE A BLOCK VARIABLE
11.          HERE SWAP , CONSTANT ;
12.
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 104(1508)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS                JV/770926   5 OF  6 )
 1. 10 MR AND,  14 MR JSB,  20 MR XOR,  24 MR JMP,  30 MR IOR,
 2. 34 MR ISZ,  40 MR ADA,  44 MR ADB,  50 MR ANE,  54 MR BNE,
 3. 60 MR LDA,  64 MR LDB,  70 MR STA,  74 MR STB,
 4. 2002 CPU A0,   6002 CPU B0,   2040 CPU E0,   2001 CPU RSS,
 5. 2020 CPU A+,   6020 CPU B+,   1727 CPU AL8,  1222 CPU 2RAL,
 6. 5727 CPU BL8,  5000 CPU BLS,  6004 CPU INB,  4022 CPU RBL,
 7. 5121 CPU 2RRS, 5222 CPU 2RBL, 2004 CPU INA,  1000 CPU ALS,
 8.   65 CPU CAER, 4065 CPU CBER, 3004 CPU TCA,  7004 CPU TCB,
 9. 2100 CPU CLE,  1600 CPU ELA,  2400 CPU CLA,  6400 CPU CLB,
10. 3000 CPU CMA,  7000 CPU CMB,  2021 CPU A-,   6021 CPU B-,
11. 1121 CPU 2ARS,
12. 27 I/O STC,  67 I/O CLC,  37 I/O STCC, 77 I/O CLCC, 20 I/O HLT,
13. 25 I/O LIA,  65 I/O LIB,  26 I/O OTA,  66 I/O OTB,  24 I/O MIA,
14. 21 I/O STF,  31 I/O CLF,  23 I/O SFS,  22 I/O SFC,  64 I/O MIB,
15. 1 +BLOCK CONTINUED

BLOCK 105(1518)
 0. ( BLOCK ASSEMBLER FOR BOOTSTRAPS                HN/780310   6 OF  6 )
 1.
 2. : DUMP           \ DUMP GENERATED CODE
 3.          BK @ BLOCK  ORIGIN @ -  DUMP-OFFSET !  \ FWA
 4.          ORIGIN @   HERE  OVER -                \ RANGE
 5.          DUMP ;
 6.
 7. CHAIN FORTH
 8. FORTH DEFINITIONS
 9. CR ." BLOCK-ASSEMBLER VOCABULARY AVAILABLE "
10. BASE ! ;S
11.
12.
13.
14.
15.
```

```
BLOCK 110(156B)
 0. ( ASSEMBLER FOR HP 21MX COMPUTERS              JV/780125   1 OF  3 )
 1. FORTH
 2.
 3. CR ." 21 MX ASSEMBLER "
 4.
 5. BASE ! OCTAL
 6. ASSEMBLER DEFINITIONS
 7.
 8. : MX CONSTANT ;CODE INB,  B ) LDA,        \ GET INSTR. CODE
 9.                     DP ) STA,  DP ISZ,    \ SHOVE IT INTO DICT.
10.                     S) LDA,               \ GET OPERAND ADDRESS
11.                     DP ) STA,  DP ISZ,    \ AND SHOVE IT
12.                     POP,
13.                     EDOC
14.
15. 1 +BLOCK CONTINUED

BLOCK 111(157B)
 0. ( ASSEMBLER FOR HP 21MX COMPUTERS              JV/771010   2 OF  3 )
 1.
 2. : RX CONSTANT ;CODE  INB,  B ) LDA,       \ GET INSTRUCTION CODE
 3.                     DP ) STA,  DP ISZ,    \ SHOVE IT
 4.                     3) LDA,               \ GET FIRST OPERAND
 5.                     DP ) STA,  DP ISZ,    \ SHOVE IT
 6.                     S1) LDA,              \ GET SECOND OPERAND
 7.                     DP ) STA,  DP ISZ,    \ SHOVE IT
 8.                     2POP,
 9.                     EDOC
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 112(160B)
 0. ( ASSEMBLER FOR HP 21MX COMPUTERS              JV/771010   3 OF  3 )
 1. 105746 MX  ADX, 105756 MX  ADY, 101741 CPU CAX, 101751 CPU CAY,
 2. 105774 RX  CBS, 105766 RX  CBT, 105741 CPU CBX, 105751 CPU CBY,
 3. 105776 RX  CMW, 101744 CPU CXA, 105744 CPU CXB, 101754 CPU CYA,
 4. 105754 CPU CYB, 105761 CPU DSX, 105771 CPU DSY, 105760 CPU ISX,
 5. 105770 CPU ISY, 105762 MX  JLY, 105772 MX  JPY, 101742 MX  LAX,
 6. 101752 MX  LAY, 105763 CPU LBT, 105742 MX  LBX, 105752 MX  LBY,
 7. 105745 MX  LDX, 105755 MX  LDY, 105765 RX  MBT, 105777 RX  MVW,
 8. 101740 MX  SAX, 101750 MX  SAY, 105773 RX  SBS, 105764 CPU SBT,
 9. 105740 MX  SBX, 105750 MX  SBY, 105767 CPU SFB, 105743 MX  STX,
10. 105753 MX  STY, 105775 RX  TBS, 101747 CPU XAX, 101757 CPU XAY,
11. 105747 CPU XBX, 105757 CPU XBY,
12.
13. FORTH DEFINITIONS
14. BASE ! ;S
15.
```

```
BLOCK 200(310B)
 0, ( BOOTSTRAP FOR MOST ROM LOADERS                HN/780319   1 OF  6 )
 1, FORTH DEFINITIONS
 2, BASE @ OCTAL
 3, : BOOTSTRAP ;
 4, BLOCK-ASSEMBLER
 5,
 6, 1 BK !                                          \ THIS BLOCK !
 7, 0 BLOCK-ADDRESS !                               \ START ON WORD 0
 8, BK @ BUFFER 0 OVER ! DUP 1+ 777 MOVE            \ ZERO BLOCK BK
 9,
10, 2011 ORIGIN !                    \ LOADER LOADS FROM HERE
11,                                  \ LOADER THEN EXECUTES JSB 2055,I
12,
13, 1 +BLOCK CONTINUED
14,
15,

BLOCK 201(311B)
 0, ( BOOTSTRAP FOR MOST ROM LOADERS                JV/780113   2 OF  6 )
 1,   2011    VAR B1A                  \ START ADDRESS OF BOOTSTRAP
 2,  16011    VAR B2A                  \ START OF MOVED BOOT
 3,  -1000    VAR B-L                  \ LENGTH OF BOOTSTRAP
 4,      0    VAR TRANSFER             \ TRANSFER ADDRESS
 5,   6011    VAR F1A                  \ START ADDRESS OF FORTH
 6,    100    VAR F2A                  \ START OF ACTUAL FORTH
 7,  -4000    VAR F-L                  \ LENGTH OF FORTH SYSTEM
 8,    100 ) VAR START                 \ FORTH COLD START ADDRESS
 9,   4000    VAR F1L                  \ LENGTH OF FORTH
10, 1 +BLOCK CONTINUED
11,
12,
13,
14,
15,

BLOCK 202(312B)
 0, ( BOOTSTRAP FOR MOST ROM LOADERS                JV/780117   3 OF  6 )
 1, 2055 HERE - DUP 0< IFTRUE ." ERROR IN BOOTSTRAP " ABORT
 2,                    IFEND
 3,                 BLOCK-ADDRESS +!
 4, 1 +BLOCK CONTINUED
 5,
 6,
 7,
 8,
 9,
10,
11,
12,
13,
14,
15,
```

```
BLOCK 203(313B)
 0. ( BOOTSTRAP FOR MOST ROM LOADERS              HN/780313   4 OF  6 )
 1. HERE 1+ ,                          \ 2055 ; JSB 2055,I
 2.       0 ,                          \ 2056 : BBDL ADDRESS
 3.                                    \ BOOTSTRAP STARTS HERE
 4. 77 HLT,                            \ 2057
 5.  0 CLC,  0 CLF,                    \ 2060
 6.  1 LIA,                            \ 2062
 7.  A0,                               \ 2063
 8.  HERE 4 + JMP,                     \ 2064
 9.  F1L LDA,                          \ 2065
10.  F1A ADA,                          \ 2066
11.  F1A STA,                          \ 2067
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 204(314B)
 0. ( BOOTSTRAP FOR MOST ROM LOADERS              HN/780313   5 OF  6 )
 1.  HERE B1A ) LDA,                   \ 2070
 2.       B2A ) STA,                   \ 2071
 3.       B1A ISZ,                     \ 2072
 4.       B2A ISZ,                     \ 2073
 5.       B-L ISZ,                     \ 2074
 6.  JMP,                              \ 2075
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 205(315B)
 0. ( BOOTSTRAP FOR MOST ROM LOADERS              HN/780313   6 OF  6 )
 1. HERE 1+ 14000 + TRANSFER !,        \ SET TRANSFER ADDRESS
 2. TRANSFER ) JMP,                    \ 2076
 3. HERE F1A ) LDA,                    \ 2077
 4.       F2A ) STA,                   \ 2100
 5.       F1A ISZ,                     \ 2101
 6.       F2A ISZ,                     \ 2102
 7.       F-L ISZ,                     \ 2103
 8. JMP,                               \ 2104
 9. 77 HLT,                            \ 2105
10. START ) JMP,                       \ 2106
11.
12. FLUSH
13. FORTH
14. BASE !
15. ;S
```

```
BLOCK 250(372B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       JV/770926   1 OF  7 )
 1. FORTH DEFINITIONS
 2. ?DEF DC 0= IFTRUE
 3.                  ." DC NOT DEFINED." CR
 4.                  ." PLEASE DEFINE DISC SELECT CODE." CR
 5.                  ABORT
 6.          IFEND
 7.
 8. CR ." BOOTSTRAP FOR 7900 DISC ROM LOADER" CR
 9.
10. DC 1+ CONSTANT CC           \ COMMAND CHANNEL SELECT CODE
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 251(373B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       HN/780313   2 OF  7 )
 1. BASE @ OCTAL
 2. BLOCK-ASSEMBLER
 3. 2 BK !                                      \ THIS BLOCK !
 4. 0 BLOCK-ADDRESS !                           \ START ON WORD 0
 5. BK @ BUFFER 0 OVER ! DUP 1+ 777 MOVE        \ ZERO BLOCK BK
 6.
 7. 2011 ORIGIN !          \ LOADER LOADS FROM HERE
 8.                        \ LOADER THEN EXECUTES: JSB 2055,I
 9.
10. \ THIS BOOTSTRAP STARTS WITH A HALT 77
11. \ THEN THE NEXT BOOTSTRAP IS LOADED  AND
12. \ A JMP 10000,I IS EXECUTED.
13.
14. 1 +BLOCK CONTINUED
15.

BLOCK 252(374B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       JV/770907   3 OF  7 )
 1.       0        VAR CYLNO    \ 2011 : CYLINDER NUMBER
 2.       4        VAR HDSEC    \ 2012 : HEAD-SECTOR ADDRESS
 3.       0        VAR DN       \ 2013 : DRIVE NUMBER
 4.   30000        VAR SKCMD    \ 2014 : SEEK COMMAND
 5.  120000  DC + VAR DCW1       \ 2015 : DMA CONTROL WORD
 6.   20000        VAR DRDCMD   \ 2016 : READ COMMAND
 7.   -1000        VAR DRSIZE   \ 2017 : WORD COUNT FOR TRANSFER
 8.  110000        VAR BUF      \ 2020 : STORE IN LOC 10000->11000
 9.  110000        VAR START    \ 2021 : START ADDRESS
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 253(375B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       JV/770926   4 OF  7 )
 1. 2055 HERE - DUP 0< IFTRUE ." ERROR IN BOOTSTRAP " ABORT
 2.                    IFEND
 3.                 BLOCK-ADDRESS +!
 4. 1 +BLOCK CONTINUED
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 254(376B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       JV/770926   5 OF  7 )
 1. HERE 1+ ,                                  \ 2055 : JSB 2055,I
 2.       0 ,                                  \ 2056 : BBDL ADDRESS
 3.                                            \ BOOTSTRAP STARTS HERE
 4. 77 HLT,                                    \ 2057
 5. CYLNO LDA,     DC OTA,     DC STCC,        \ 2060
 6. SKCMD LDA,     DN IOR,     CC OTA,         \ 2063
 7.    CC STCC,                                \ 2066
 8. HERE           DC SFS,        JMP,         \ 2067
 9. HDSEC LDA,     DC OTA,     DC STCC,        \ 2071
10. HERE           CC SFS,        JMP,         \ 2074
11.    DC CLC,     CC CLC,                     \ 2076
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 255(377B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER       JV/770926   6 OF  7 )
 1.    DN LDB,     CC OTB,     CC STCC,        \ 2100
 2. HERE           DC SFS,        JMP,         \ 2103
 3.    DC LIB,     4010 ( SLB ERROR? ) ,       \ 2105
 4.    44 HLT,          ( ERROR )              \ 2107
 5.    DC CLC,     CC CLC,                     \ 2110
 6. 1 +BLOCK CONTINUED
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 256(400B)
 0. ( BOOTSTRAP FOR 7900 DISC ROM LOADER      JV/770926   7 OF  7 )
 1.  DCW1 LDA,     6 OTA,                \ 2112
 2.    BUF LDA,    2 CLC,     2 OTA,     \ 2114
 3.      2 STC, ORSIZE LDA,   2 OTA,     \ 2117
 4. DRDCMD LDB,   DN ADB,    CC OTB,     \ 2122
 5.     DC STCC,   6 STCC,   CC STCC,    \ 2125
 6. HERE          CC SFS,      JMP,      \ 2130
 7. START ) JMP,                         \ 2132
 8.
 9. FLUSH
10. FORTH
11. BASE !
12. ;S
13.
14.
15.

BLOCK 260(404B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH           JV/770926   1 OF  7 )
 1. FORTH DEFINITIONS
 2. ?DFF DC @= IFTRUE
 3.                  ." DC NOT DEFINED." CR
 4.                  ." PLEASE DEFINE DISC SELECT CODE." CR
 5.                  ABORT
 6.            IFEND
 7.
 8. CR ." BOOTSTRAP FOR 7900 DISC FORTH" CR
 9.
10. DC 1+ CONSTANT CC          \ COMMAND CHANNEL SELECT CODE
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 261(405B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH           HN/780313   2 OF  7 )
 1. BASE @ OCTAL
 2. BLOCK-ASSEMBLER
 3. 2 BK !                                    \ THIS BLOCK !
 4. 0 BLOCK-ADDRESS !                         \ START ON WORD 0
 5. BK @ BUFFER 0 OVER ! DUP 1+ 777 MOVE      \ ZERO BLOCK BK
 6.
 7. 10000 ORIGIN !          \ LOADER LOADS FROM HERE
 8.                         \ LOADER THEN EXECUTES: JMP 10000,I
 9.
10. \ THIS BOOTSTRAP STARTS WITH A HALT 77
11. \ THEN IT WILL LOAD 7900 DISC FORTH  AND
12. \ EXECUTE A JMP 100,I
13.
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 262(406B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH                JV/771018   3 OF  7 )
 1.        0          VAR BEGIN    \ 10000 : TRANSFER ADDRESS
 2.        0          VAR CYLNO    \ 10001 : CYLINDER NUMBER
 3.       20          VAR HDSEC    \ 10002 : HEAD-SECTOR ADDRESS
 4.        0          VAR DN       \ 10003 : DRIVE NUMBER
 5.    30000          VAR SKCMD    \ 10004 : SEEK COMMAND
 6.   120000  DC +     VAR DCW1     \ 10005 : DMA CONTROL WORD
 7.    20000          VAR DRDCMD   \ 10006 : READ COMMAND
 8.    -3000          VAR DRSIZE   \ 10007 : WORD COUNT FOR TRANSFER
 9.   100100          VAR BUF      \ 10010 : STORE IN LOC 100->3100
10.   100100          VAR START    \ 10011 : START ADDRESS
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 263(407B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH                JV/770926   4 OF  7 )
 1.
 2. HERE BEGIN !,                \ BOOTSTRAP STARTS HERE
 3.
 4. 1 +BLOCK CONTINUED
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 264(410B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH                JV/770926   5 OF  7 )
 1. 77 HLT,                                       \ 10012
 2. CYLNO LDA,     DC OTA,     DC STCC,           \ 10013
 3. SKCMD LDA,     DN IOR,     CC OTA,            \ 10016
 4.    CC STCC,                                   \ 10021
 5. HERE           DC SFS,        JMP,            \ 10022
 6. HDSEC LDA,     DC OTA,     DC STCC,           \ 10024
 7. HERE           CC SFS,        JMP,            \ 10027
 8.    DC CLC,     CC CLC,                        \ 10031
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 265(4115)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH                JV/770926   6 OF  7 )
 1.    DN LDB,     CC OTB,     CC STCC,  \ 10033
 2. HERE           DC SFS,        JMP,   \ 10036
 3.    DC LIB,     4010 ( SLB ERROR? ) , \ 10040
 4.    44 HLT,          ( ERROR )        \ 10042
 5.    DC CLC,     CC CLC,               \ 10043
 6. 1 +BLOCK CONTINUED
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 266(412B)
 0. ( BOOTSTRAP FOR 7900 DISC FORTH                JV/770926   7 OF  7 )
 1.  DCW1 LDA,     6 OTA,                 \ 10045
 2.   BUF LDA,     2 CLC,     2 OTA,      \ 10047
 3.     2 STC, DRSIZE LDA,    2 OTA,      \ 10052
 4. DRDCMD LDB,   DN ADB,    CC OTB,      \ 10055
 5.    DC STCC,   6 STCC,    CC STCC,     \ 10060
 6. HERE           CC SFS,        JMP,   \ 10063
 7. START ) JMP,                         \ 10065
 8.
 9. FLUSH
10. FORTH
11. BASE !
12. ;S
13.
14.
15.

BLOCK 300(454B)
 0. ( METAFORTH INITIALIZATION                     JV/770926   1 OF ** )
 1. 0 WARNING !          \ NO WARNING MESSAGES
 2. FORTH DEFINITIONS
 3. VOCABULARY METAFORTH     METAFORTH DEFINITIONS
 4.
 5. CR ." HP METAFORTH" CR
 6.
 7. OCTAL
 8. 12 +BLOCK LOADER GENERATION
 9.
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 301(455B)
 0. ( METAFORTH INITIALIZATION                    JV/780330   2 OF ** )
 1.       0 CONSTANT LOG              \ LOG/NOLOG SWITCH
 2.      11 CONSTANT SAVE-AREA        \ BLOCK # FOR SAVE ROUTINE
 3.       4 CONSTANT SAVE-#-BLOCKS    \ # BLOCKS TO BE SAVED
 4.    3777 CONSTANT BLENGTH          \ # WORDS FOR GENERATION
 5.       3 CONSTANT #CHF             \ # OF CHARACTERS IN ENTRY
 6.   60000 CONSTANT MEMORY           \ MEMORY SIZE
 7.      50 CONSTANT RP-LENGTH        \ RETURN STACK LENGTH
 8.     100 CONSTANT ORG              \ ORIGIN OF METAFORTH CODE
 9.     100 CONSTANT BBLL             \ LENGTH OF LOADER
10.    1000 CONSTANT BLOCK-LENGTH     \ LENGTH OF FORTH BLOCKS
11.       2 CONSTANT #-BUFFERS        \ NUMBER OF BUFFERS
12.    1000 CONSTANT MAP-LENGTH       \ LENGTH OF TAPE MAP
13.     200 CONSTANT #-LENGTH         \ BASE PAGE CONSTANT AREA LENGTH
14. 1 +BLOCK CONTINUED
15.

BLOCK 302(456B)
 0. ( METAFORTH INITIALIZATION                    JV/780313   3 OF ** )
 1.      24 CONSTANT U1#              \ LENGTH USER AREA NEEDING
 2.                                   \ SAVE/RESTORE
 3.      24 CONSTANT U2#              \ LEGTH INDEXED USER AREA
 4.      12 CONSTANT DC               \ DISC DATA CHANNEL
 5.   DC 1+ CONSTANT CC               \ DISC COMMAND CHANNEL
 6.      14 CONSTANT TTY              \ CONSOLE SELECT CODE
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 303(457B)
 0. ( METAFORTH INITIALIZATION                    HN/780212   4 OF ** )
 1.       3 CONSTANT COMPUTER          \ SPECIFIES COMPUTER TYPE
 2. \ BIT  0 : DMA AVAILABLE
 3. \ BIT  1 : HARDWARE MULTIPLY AND DIVIDE ( EAU )
 4. \ BIT 15 : INTERRUPT DRIVERS WHEN AVAILABLE
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 304(460B)
 0. ( METAFORTH INITIALIZATION                          JV/780403   5 OF ** )
 1.       3 CONSTANT MASS-STORAGE  \ DEFINES MASS STORAGE DEVICE
 2. \ BIT 0-7 :   MASS-STORAGE DEVICE
 3. \             1     HP 7900 DISC PLATTER
 4. \             3     HP 7900 DISC CONTINUED PLATTERS
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 305(461B)
 0. ( METAFORTH INITIALIZATION                          HN/780313   6 OF ** )
 1.
 2. : SET  'C, ' ! ;  \ FOR EASY CHANGING OF GENERATION CONSTANTS
 3.
 4. CR CR
 5. ." PLEASE DEFINE NON-STANDARD GENERATION CONSTANTS" CR
 6. ." THEN TYPE 'GENERATION' TO GENERATE NEW SYSTEM" CR
 7. ." NEW SYSTEM WILL BE WRITTEN TO MASS STORAGE DEVICE WITH" CR
 8. ." 'SAVE'" CR
 9. CR ." NEW SYSTEM WILL BE EXECUTED WITH 'TRY' "  CR
10. 1 WARNING !
11. ;S
12.
13.
14.
15.

BLOCK 310(466B)
 0. ( METAFORTH GENERATION CONSTANTS                    HN/780315   1 OF ** )
 1. OCTAL  0 WARNING !  METAFORTH DEFINITIONS
 2. CR CR CR CR ." METAFORTH GENERATION CONSTANTS" CR CR
 3. LOG IFTRUE ." SYSTEM ENTRIES WILL BE LOGGED" CR CR IFEND
 4. ." SYSTEM SAVING BLOCKS       " SAVE-AREA . ." THRU "
 5.                                 SAVE-AREA SAVE-#-BLOCKS + 1- .
 6. CR
 7. ." CHARACTERS IN ENTRY       "
 8.   #CHF IFTRUE  #CHF . OTHERWISE ." FLOATING" IFEND CR
 9.   #CHF 1+ 2/ CONSTANT #WDS
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 311(467B)
 0. ( METAFORTH GENERATION CONSTANTS              HN/780313   2 OF ** )
 1. ." MEMORY SIZE                    " MEMORY 0. CR
 2. ." RETURN STACK LENGTH            " RP-LENGTH 0. CR
 3. ." ORIGIN OF METAFORTH CODE       " ORG 0. CR
 4. ." LOADER LENGTH                  " BBLL 0. CR
 5. ." FORTH BLOCK LENGTH             " BLOCK-LENGTH 0. CR
 6. ." NUMBER OF BUFFERS              " #-BUFFERS 0. CR
 7. ." MAP LENGTH ( OPTIONAL )        " MAP-LENGTH 0. CR
 8. ." BASE PAGE CONSTANTS            " #-LENGTH 0. CR
 9. ." USER AREA LENGTH 'S/R'         " U1#  0. CR
10. ." USER AREA LENGTH 'IND'         " U2#  0. CR
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 312(470B)
 0. ( METAFORTH GENERATION CONSTANTS              JV/780313   3 OF ** )
 1. CR CR ." SYSTEM SELECT CODES" CR
 2. ." TTY CONSOLE                    " TTY 0. CR
 3. MASS-STORAGE 377 AND DUP 1 = SWAP 3 = OR
 4.                      IFTRUE ." 7900 DISC SELECT CODES:" DC .
 5.                             CC . CR
 6.                      IFEND
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 313(471B)
 0. ( METAFORTH GENERATION CONSTANTS              HN/780313   4 OF ** )
 1. MEMORY BBLL - SET MEMORY                    \ UPDATE MEMORY SIZE
 2.
 3. 1 +BLOCK CONTINUED
 4.
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 314(472B)
 0. ( METAFORTH GENERATION CONSTANTS          JV/771114   5 OF ** )
 1. CR CR ." COMPUTER SPECIFICATION " COMPUTER 0. CR
 2. COMPUTER 100000 AND IFTRUE ." INTERRUPT DRIVERS WILL BE "
 3.                            ." LOADED WHEN AVAILABLE" CR  IFEND
 4. COMPUTER 000001 AND IFTRUE ." DMA AVAILABLE" CR          IFEND
 5. COMPUTER 000002 AND IFTRUE ." HARDWARE MPY, DIV," CR     IFEND
 6. 1 +BLOCK CONTINUED
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 315(473B)
 0. ( METAFORTH GENERATION CONSTANTS          HN/780212   6 OF ** )
 1. CR CR ." MASS STORAGE SPECIFICATION " MASS-STORAGE 0. CR
 2. MASS-STORAGE  377 AND   1 = IFTRUE ." HP 7900 DISC " CR IFEND
 3. MASS-STORAGE  377 AND   3 = IFTRUE ." HP 7900 DISC ,"
 4.                                    ." CONTINUED PLATTERS " CR
 5.                                    IFEND
 6. 5 +BLOCK ( 320D ) CONTINUED
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 320(500B)
 0. ( METAFORTH BASIC GENERATION WORDS        JV/780315   1 OF ** )
 1. CR CR ." METAFORTH BASIC GENERATION WORDS"
 2.     0 VARIABLE BUF                  \ FIRST WORD OF METAFORTH BUF.
 3.               BLENGTH DP+!          \ GENERATION BUFFER LENGTH
 4. BUF BUF 1+ BLENGTH MOVE             \ ZERO METAFORTH BUFFER
 5. ORG VARIABLE DP                     \ DICTIONARY POINTER
 6. : HERE DP @ ;                       \ PUSH DICTIONARY POINTER
 7. : CP   ORG - 0 MAX BLENGTH MIN BUF + ! \ MAP ADDRESS INTO BUFFER
 8. : DUMP ORG CP ORG - DUMP-OFFSET ! OVER - DUMP ; \ DUMP BUFFER
 9. : SAVE SAVE-#-BLOCKS 0              \ RANGE
10.         DO BUF I 1000 * +           \ SOURCE ADDRESS
11.            SAVE-AREA I + BUFFER     \ DESTINATION BLOCK
12.            1000 MOVE                \ MOVE THE WORDS
13.         LOOP
14.         FLUSH ;
15. 1 +BLOCK CONTINUED
```

```
BLOCK 321(5018)
 0. ( METAFORTH BASIC GENERATION WORDS          JV/771018   2 OF ** )
 1. 4 ,CODE TRY  BUF ,   ORG ,   -4000 ,   ORG ) ,
 2.              FORTH HERE METAFORTH ASSEMBLER
 3.                    ' TRY ) LDA,  ' TRY ISZ,   \ FETCH WORD
 4.                    ' TRY 1+ ) STA,            \ STORE WORD
 5.                    ' TRY 1+ ISZ,              \ INCREMENT ADDRESS
 6.                    ' TRY 2 + ISZ,             \ READY?
 7.              JMP,                             \ NO
 8.              77 HLT,                          \ YES.
 9.              ' TRY 3 + ) JMP,                 \ JUMP!
10. EDOC
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 322(5028)
 0. ( METAFORTH BASIC GENERATION WORDS          JV/780313   3 OF ** )
 1. : TEXT   LAST @ ;                 \ LEAVE ADDRESS OF LAST ENTRY
 2. : +COUNT FORTH #CHF
 3.          IF   #CHF
 4.          ELSE HERE COUNT SWAP DROP
 5.          THEN
 6.          2/ 1+ 2 MAX
 7.          METAFORTH ;              \ LEAVE WORD COUNT OF ENTRY ID.
 8. : RENAME 40 WORD FORTH HERE METAFORTH TEXT +COUNT MOVE ;
 9.                                   \ RENAME LAST ENTRY IN DICTIONARY
10. : '      'C, ' @ ;                \ ' LEAVES METAFORTH ADDRESS
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 323(5038)
 0. ( METAFORTH BASIC GENERATION WORDS          JV/780321   4 OF ** )
 1. : LOG            \ LOG ALL ENTRIES ACCORDING TO LOG SWITCH
 2.       IN @ 40 WORD IN !                    \ ALWAYS READ A WORD
 3.       LOG IF   \ SWITCH SET?
 4.                CR                         \ NEW LINE
 5.                FORTH HERE METAFORTH       \ LOG ID TEXT ON DICTIONARY
 6.                COUNT SWAP OVER            \ CHARACTER COUNT
 7.                #CHF IF #CHF MIN THEN      \ MAY NOT EXCEED #CHF
 8.                WRITE SPACE .              \ PRINT IDENTIFICATION
 9.                HERE .                     \ PRINT ADDRESS OF ENTRY
10.                FORTH HERE .               \ DICTIONARY ENTRY ADDRESS
11.                METAFORTH
12.           THEN ;
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 324(504B)
 0. ( METAFORTH BASIC GENERATION WORDS          HN/780212   5 OF ** )
 1. : !      CP ! ;                          \ ! MAPS IN BUFFER
 2. : DP+!   DP +! ;                         \ INCREMENT DP
 3. : ,      HERE ! 1 DP+! ;                 \ COMPILES IN BUFFER
 4. : +!     CP +! ;                         \ +! MAPS IN BUFFER
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 325(505B)
 0. ( METAFORTH BASIC GENERATION WORDS          JV/780118   6 OF ** )
 1.    0 VARIABLE HEAD                        \ HEAD OF VOCABULARY
 2.    0 VARIABLE OTHER                       \ HEAD OF ALTERNATE
 3.                                           \ VOCABULARY
 4.    0 VARIABLE ADR                         \ DEFAULT CODE ADDRESS
 5.
 6. : ASSEMBLER-DEFINITIONS HEAD @ OTHER      \ HEAD OF VOCABULARY
 7.                         OTHER @ HEAD      \ ALTERNATE VOCABULARY
 8.                         FORTH ! ! METAFORTH ;   \ EXCHANGE BOTH
 9. :    FORTH-DEFINITIONS ASSEMBLER-DEFINITIONS ; \ EXCHANGE BOTH
10.
11. : OCT       LOG HERE CONSTANT , ;        \ LABEL A LOCATION
12. : USER      LOG CONSTANT ;               \ DEFINE USER VARIABLE
13. : UABS      DROP HERE CONSTANT ;         \ ABSOLUTE USER ADDRES
14. : COMPILE   ?: , ;                       \ REFERENCE BY ADDRESS
15. 1 +BLOCK CONTINUED

BLOCK 326(506B)
 0. ( METAFORTH BASIC GENERATION WORDS          HN/780313   7 OF ** )
 1. : ENTRY      HERE                         \ ADDRESS FOR VOCABULARY LINK
 2.              FORTH 0 HERE ! HERE DUP 1+   \ ZERO TOP OF DICT.
 3.              METAFORTH #WDS 2 MAX 1- MOVE \ BEFORE READING
 4.              LOG                          \ READ WORD AND LOG IT
 5.              FORTH HERE METAFORTH         \ WORD STARTS HERE
 6.              #CHF IF #WDS                 \ FIXED ENTRY
 7.                ELSE DUP COUNT SWAP DROP   \ FLOATING ENTRY
 8.                     2/ 1+                 \ COUNT->WORDS
 9.                THEN 2 MAX OVER + SWAP     \ RANGE TO BE MOVED
10.              DO I @ 77577 AND , LOOP      \ CONSTRUCT TEXT ENTRY
11.              HEAD @ ,                     \ SET LINK FIELD
12.              HEAD FORTH ! METAFORTH       \ UPDATE HEAD ADDRESS
13.              HERE COMPILE                 \ REFERENCE BY ADDRESS
14.              ADR @ , ;                    \ DEFAULT CODE ADDRESS
15. 1 +BLOCK CONTINUED
```

```
BLOCK 327(5078)
 0. ( METAFORTH BASIC GENERATION WORDS          HN/770929   8 OF ** )
 1. : IMMEDIATE 100000 HERE 3 - +! ;            \ SET PRECEDENCE BIT
 2. : ADDRESS   HERE 1- ;                       \ CODE FIELD ADDRESS
 3. : ,CODE     ENTRY HERE + ADDRESS ! ;        \ ,CODE ENTRY
 4. : CODE      0 ,CODE ;                       \ CODE  ENTRY
 5. : <CODE     LOG  HERE COMPILE               \ LOG 'ENTRY'
 6.             HERE 1+ , ;                     \ SET CODE ADDRESS
 7. : <:        LOG  HERE COMPILE               \ LOG 'ENTRY'
 8.             ADR @ , ;                       \ SET CODE ADDRESS
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 328(5108)
 0. ( METAFORTH BASIC GENERATION WORDS          JV/780321   9 OF ** )
 1. 0  VARIABLE  1CON
 2. 0  VARIABLE  1VAR
 3. 0  VARIABLE  1CPU
 4. : CON       ENTRY  1CON @  ADDRESS  !  ;
 5. : VAR       ENTRY  1VAR @  ADDRESS  !  ;
 6. : 2CPU      ENTRY  1CPU @  ADDRESS  !  ;
 7. : PASS      IN @ CON                        \ CONSTRUCT ENTRY
 8.             0 TEXT FORTH ! METAFORTH        \ DESTROY FORTH ENTRY
 9.             IN ! ;                          \ EXECUTE WORD AGAIN
10.
11. 2 +BLOCK ( 3300 ) CONTINUED
12.
13.
14.
15.

BLOCK 330(5128)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  HN/780319   1 OF ** )
 1.    0 VARIABLE #1                           \ START OF CONSTANT AREA
 2.    0 VARIABLE #2                           \ POINTER IN AREA
 3. : #      1 SWAP  #2 @  #1 @                \ DO FOR OCCUPIED AREA
 4.          DO   I CP @ OVER =                \ FOUND ?
 5.               IF  2DROP 0 I LEAVE THEN     \ YES, IT EXISTS
 6.          LOOP
 7.          SWAP                              \ IS IT ALREADY FOUND?
 8.          IF   #2 @ !  #2 @                 \ STORE IN AREA
 9.               1 #2 FORTH +! METAFORTH      \ INCREMENT POINTER
10.               #2 @  #1 @ - #-LENGTH >      \ TABLE OVERFLOW?
11.               IF ." CONSTANT AREA OVERFLOW" ABORT THEN
12.          THEN ;
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 331(5135)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  JV/771021   2 OF ** )
 1. : CPU        COMPILE ;                          \ CPU INSTRUCTION
 2. : )          100000 + ;                         \ SET INDIRECT BIT
 3. : AEIOU      100 * ) ;                          \ FOR AEIOU GROUP
 4. : I/O        AEIOU ;: + , ;                     \ I/O INSTRUCTION
 5. : DEAU       AEIOU ;: , , ;                     \ DEAU GROUP
 6. : MR         1000 * ;:                          \ MEMORY REFERENCE
 7.              SWAP  DUP 101777 AND OVER =  \ ZERO PAGE?
 8.              IF  DUP HERE XOR 76000 AND   \ CURRENT PAGE?
 9.                  IF  # )  THEN            \ GENERATE LINK
10.              THEN
11.              * , ;                              \ ASSEMBLE INSTR.
12.   0 CONSTANT A                                  \ A REGISTER
13.   1 CONSTANT B                                  \ B REGISTER
14. 1 +BLOCK CONTINUED
15.

BLOCK 332(5146)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  JV/780330   3 OF ** )
 1. 10 MR AND,  14 MR JSB,  20 MR XOR,  24 MR JMP,  30 MR IOR,
 2. 34 MR ISZ,  40 MR ADA,  44 MR ADB,  50 MR ANE,  54 MR BNE,
 3. 60 MR LDA,  64 MR LDB,  70 MR STA,  74 MR STB,
 4. 2002 CPU A0,   6002 CPU B0,   2040 CPU E0,   2001 CPU RSS,
 5. 2020 CPU A+,   6020 CPU B+,   1727 CPU AL8,  1222 CPU 2RAL,
 6. 5727 CPU BL8,  5000 CPU BLS,  6004 CPU INB,  4022 CPU RBL,
 7. 5121 CPU 2RRS, 5222 CPU 2RBL, 2004 CPU INA,  1000 CPU ALS,
 8.   65 CPU CAER, 4065 CPU CBER, 3004 CPU TCA,  7004 CPU TCB,
 9. 2100 CPU CLE,  1600 CPU ELA,  2400 CPU CLA,  6400 CPU CLB,
10. 3000 CPU CMA,  7000 CPU CMB,  2021 CPU A-,   6021 CPU B-,
11. 1121 CPU 2ARS,
12. 27 I/O STC,  67 I/O CLC,  20 I/O HLT,  25 I/O LIA,  65 I/O LIB,
13. 26 I/O OTA,  66 I/O OTB,  24 I/O MIA,  21 I/O STF,  31 I/O CLF,
14. 23 I/O SFS,  22 I/O SFC,  64 I/O MIB,
15. 1 +BLOCK CONTINUED

BLOCK 333(5158)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  JV/770926   4 OF ** )
 1. : DIRECT     HERE ;: JMP, ;                    \ DIRECT TRANSFER
 2. : RETURN     ) JMP, ;                          \ SUBROUTINE RETURN
 3. : INDIRECT  ;: RETURN ;                        \ INDIRECT TRANSFER
 4. : SUBROUTINE HERE 0 , DUP ;: JSB, ;    \ SUBROUTINE DEF.
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 334(516B)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  JV/780330   5 OF ** )
 1. : NOT        HERE 1- DUP CP @ 1          \ ENABLE RSS INSTR.
 2.              XOR  SWAP  !  ;
 3. : CLF        HERE 1- DUP CP @ 1000
 4.              OR  SWAP  !  ;
 5. : SUB,       TCA, ADA, TCA, ;            \ SUBTRACT MACRO
 6. : ADM,       DUP  ADA,  STA,  ;          \ ADD-M MACRO
 7. : IF,        HERE 0 , ;                  \ 'IF' MACRO
 8. : THEN,      HERE SWAP DP FORTH !        \ RESET DP
 9.              METAFORTH DUP  JMP,         \ GENERATE JMP,
10.              DP FORTH ! METAFORTH ;      \ RESTORE DP
11. : ELSE,      IF, SWAP THEN, ;            \ 'ELSE' MACRO
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 335(517B)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  HN/780212   6 OF ** )
 1. COMPUTER 000002 AND          \ HARDWARE MPY AND DIV
 2.                  IFTRUE 2 DEAU MPY,
 3.                         4 DEAU DIV,
 4.                  IFEND
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 336(520B)
 0. ( METAFORTH ASSEMBLER FOR HP 2100 SERIES  HN/780313   7 OF ** )
 1. ORG 1+ CONSTANT SP                       \ STACK POINTER
 2.  SP 1+ CONSTANT SP1
 3.  SP ) CONSTANT S)                        \ TOS
 4.  SP1 ) CONSTANT S1)                      \ TOS-1
 5. : ISP,       SP ISZ, SP1 ISZ, ;          \ INCREMENT SP
 6. : DSP,       SP LDB, SP1 STB, -1 # ADB,  \ DECREMENT SP
 7.              SP STB, ;
 8. : :          ENTRY ;                     \ :-ENTRY
 9.
10. 4 +BLOCK ( 340D ) CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 340(524B)
 0. ( METAFORTH GENERATION                                HN/780315   1 OF ** )
 1. 0 OCT COLD-START                                     \ COLD START ADDRESS
 2. 2 DP+!                                               \ 101 : SP
 3.                                                      \ 102 : SP1
 4. 0 OCT GO                                             \ IC OF INTERPRETER LOOP
 5. 0 OCT N    0 ,     \ TEMP. STORAGE, USED BY NUMBER AND FIND
 6. 0 OCT CH                                             \ CHARACTER ( ADDRESS )
 7. 0 OCT BASE0                                          \ TEMP. CONVERSION BASE
 8. 0 OCT SIGN                                           \ SIGN OF NUMBER
 9. HERE INDIRECT ABORT,  0 ,                            \ ABORT ADDRESS
10. HERE INDIRECT ERROR,  0 ,                            \ ERROR ADDRESS
11. 2 CONSTANT U                                         \   2 : USER ADDRESS
12. 3 CONSTANT OWNER                                     \   3 : MASS-STORAGE
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 341(525B)
 0. ( METAFORTH GENERATION                                HN/780319   2 OF ** )
 1. COMPUTER 000002 AND 0=                    \ NO HARDWARE MPY,DIV?
 2.                  IFTRUE 0 OCT MULTIPLY-DIVIDE  0 ,
 3.                         FORTH : MPY, MULTIPLY-DIVIDE    ) JSB,
 4.                                       , ;
 5.                         FORTH : DIV, MULTIPLY-DIVIDE 1+ ) JSB,
 6.                                       , ;
 7. \ REMARK : FORTH AUTOMATICALLY SWITCHES TO METAFORTH
 8.                  IFEND
 9.
10. 4 +BLOCK ( 345D ) CONTINUED
11.
12.
13.
14.
15.

BLOCK 345(531B)
 0. ( METAFORTH MONITOR                                   JV/771103   1 OF  1 )
 1. CR ." WAIT ROUTINE " HERE .
 2. HERE  1+ OCT SUSPEND                                 \ PAUSE ROUTINE
 3. SUSPEND ) 14000 + CPU PAUSE,
 4.
 5. \ PAUSE, DESTROYS ALL REGISTERS
 6.
 7. 0 ,                                                  \ FOR RETURN ADDRESS
 8. HERE   U ) LDA,  A0,   JMP,                          \ WAIT UNTIL STATUS 0
 9. SUSPEND 1+ ) JMP,                                    \ RETURN
10.
11. 5 +BLOCK ( 350D ) CONTINUED
12.
13.
14.
15.
```

```
BLOCK 350(536B)
 0. ( METAFORTH CONSTANT AREA                          HN/780319   1 OF  1 )
 1. CR ." CONSTANT AREA STARTING AT " HERE .
 2.
 3. HERE          #1 FORTH ! METAFORTH    \ FIRST WORD OF AREA
 4. HERE          #2 FORTH ! METAFORTH    \ POINTER IN AREA
 5. #-LENGTH DP+!                         \ RESERVE AREA
 6.
 7. 177777              # DROP            \ FIRST WORD IN AREA
 8.
 9. 5 +BLOCK ( 355D ) CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 355(543B)
 0. ( METAFORTH USER AREA                              JV/780118   1 OF ** )
 1. CR ." USER AREA ALLOCATION " HERE .
 2. HERE CONSTANT USER-AREA
 3.     0 USER STATUS                    0 , \ USER STATUS
 4.     1 USER FOLLOWER          USER-AREA , \ ADDRESS OF NEXT USER
 5.     2 USER RETURN-ADDRESS            0 , \ FROM ASSEMBLER PROGS
 6.     3 USER TIME-OUT                  0 , \ TIME OUT VALUE
 7.     4 USER STACK-POINTER             0 ,
 8.     5 UARS RP                        0 ,
 9.     6 UARS SP0                       0 ,
10.     7 UARS RP0                       0 ,
11.    10 UARS DP                        0 ,
12.    11 UARS STATE                     0 ,
13.    12 UARS CURRENT                   0 ,
14.    13 UARS CONTEXT                   0 ,
15. 1 +BLOCK CONTINUED

BLOCK 356(544B)
 0. ( METAFORTH USER AREA                              HN/780313   2 OF ** )
 1.    14 UARS LAST                      0 ,
 2.    15 UARS OP                        0 ,
 3.    16 UARS IP                        0 ,
 4.    17 UARS IN                        0 ,
 5.    20 UARS SCR                       0 ,
 6.    21 UARS MSG                       0 ,
 7.    22 UARS DPL                       0 ,
 8.    23 UARS BASE                     10 ,
 9.    24 USER ASK                       0 ,
10.    25 USER TALK                      0 ,
11.    26 USER VOC-LINK                  0 ,
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 357(545B)
 0. ( METAFORTH USER AREA                                JV/780201   3 OF ** )
 1. U1# U2# +  HERE USER-AREA - - DP+!   \ INCREMENT DP
 2. 10 +BLOCK ( 365D ) CONTINUED
 3.
 4.
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 365(555B)
 0. ( METAFORTH INTERPRETER                              HN/780319   1 OF ** )
 1. CR ." INTERPRETER " HERE .
 2.
 3. CODE    LIT          RP ) LDA,  A ) LDA,    \ PUSH LITERAL
 4. DIRECT  PUSH-SKIP    RP ) TSZ,              \ INCREMENT IC
 5. DIRECT  PUSH,        DSP,                   \ RESERVE LOC.
 6. DIRECT  PUT,         S) STA,                \ PUT VALUE ON STACK
 7. DIRECT  NEXT,        RP ) LDB,  B ) LDB,    \ CONTENTS IC
 8.                      RP ) ISZ,              \ INCREMENT IC
 9. DIRECT  ,NEXT,       B ) LDA,  A ) JMP,     \ EXECUTE NEXT WORD
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 366(556B)
 0. ( METAFORTH INTERPRETER                              HN/780319   2 OF ** )
 1. <CODE  EXECUTE \ EXECUTE THE WORD JUST FOUND, ADDRESS IN N+1
 2.                N 1+ LDB,                    \ WORD ADDRESS
 3.              #CHF IFTRUE #WDS 1- # ADB,     \ ADDRESS OF PRECEDENCE
 4.                   OTHERWISE B ) LDA, AL8,   \ COUNT IN LOW BYTE
 5.                             377 # AND,      \ CHARACTER COUNT
 6.                             21 ( ARS ) ,    \ #-WORDS-1
 7.                             A0, NOT  INA,   \           >=2
 8.                             A ADB,          \ PRECEDENCE ADDRESS
 9.                   IFEND
10.               B ) LDA,  0 ) # AND,          \ ELIMINATE PRECEDENCE
11.               2 # ADB,  A-,                 \ PRECEDENCE?
12.               STATE ANE,                    \ IF PRECEDENCE OR
13.               ,NEXT,                        \ IMMEDIATE MODE
14.                                             \ EXECUTE NEXT WORD ;
15. 1 +BLOCK CONTINUED
```

```
BLOCK 367(557B)
 0. ( METAFORTH INTERPRETER                          HN/780319   3 OF ** )
 1. DIRECT SHOVE   DP ) STB, DP ISZ,                 \ ELSE STORE IN DIC.
 2.                CP LDA, CMA, SP ADA,              \ DICT. OVERFLOW?
 3.                A=,                               \ OVERFLOW?
 4.                NEXT,                             \ NO, CONTINUE
 5.                1 # LDA,                          \ YES, ABORT
 6.                ERROR,
 7.
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 368(560B)
 0. ( METAFORTH INTERPRETER                          JV/780320   4 OF ** )
 1. DIRECT 2POP,   ISP,                              \ INCREMENT SP
 2. DIRECT  POP,   SP LDA,                           \ TEST UNDERFLOW
 3.                ISP,                              \ INCREMENT SP
 4. DIRECT  STACK? CMA,  SP0 ADA,
 5.                A=,                               \ UNDERFLOW?
 6.                NEXT,                             \ NO, CONTINUE
 7.                2 # LDA,                          \ YES, ABORT
 8.                ERROR,
 9. ( BINARY, )    ISP,                              \ INCREMENT SP
10. DIRECT BINARY, ISP,  S) STA,                     \ INCREMENT SP, PUT
11.                SP LDA,                           \ TEST UNDERFLOW
12.                STACK?                            \ TEST
13. 2 +BLOCK ( 370D ) CONTINUED
14.
15.

BLOCK 370(562B)
 0. ( METAFORTH BASIC VOCABULARY                     HN/780313   1 OF ** )
 1. CR ." START OF BASIC-FORTH VOCABULARY " HERE .
 2.
 3. ENTRY ASSEMBLER IMMEDIATE                        \ CONSTRUCT ENTRY
 4.                1600 ,                            \ PSEUDO ENTRY HEAD
 5.             HERE OTHER FORTH ! METAFORTH
 6.                1600 ,                            \ PSEUDO ENTRY LINK
 7.          #WDS 2 - 0 MAX  DP+!                     \ COMPLETE ENTRY
 8.                   0 ,                            \ HEAD OF VOCABULARY
 9.                   0 ,                            \ LINK OF VOCABULARY
10.                   0 ,                            \ VOCABULARY CHAIN
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.
```

```
BLOCK 371(563B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780313   2 OF ** )
 1.
 2.  HERE ( ;CODE )                                 \ COMPILE ;CODE
 3.    #WDS 2 MAX  ' ASSEMBLER 1+ + # LDA,          \ HEAD OF ASSEMBLER
 4.                CONTEXT STA,                     \ CONTEXT TO ASSEMBLER
 5.
 6.  HERE ( ; )      LAST LDA,                      \ COMPILE ;
 7.                  CURRENT ) STA,                 \ INCLUDE ENTRY IN DIC.
 8.                  CLA,  STATE STA,               \ IMMEDIATE MODE
 9.                  INB,  SHOVE                    \ COMPILE ';' OR ';CODE'
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 372(564B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780317   3 OF ** )
 1. CODE   ;       IMMEDIATE ADDRESS !              \ ; ENTRY
 2. <CODE  ;                                        \ ; EXECUTION
 3.                HERE 1-                          \ TO CORRECT LINKAGE
 4. CODE   ;CODE   IMMEDIATE SWAP ADDRESS !         \ ;CODE ENTRY
 5. <CODE  ;CODE                                    \ ;CODE EXECUTION
 6.                LAST LDB,                        \ SET ADDR. FIELD TO IC
 7.             #CHF IFTRUE #WDS 1+ # ADB,          \ ADDR. FIELD LAST DEF
 8.                  OTHERWISE B ) LDA, ALB,       \ COUNT OF LAST DEF.
 9.                                377 # AND, 21 ( ARS ) ,  \ COUNT/2
10.                                A0, NOT  INA,    \ COUNT >= 2
11.                                2 # ADA, A ADB,  \ ADDR. FIELD LAST DEF
12.                  IFEND
13.                RP ) LDA,                        \ GET CODE ADDRESS
14.                B ) STA,                         \ SET REFERENCE
15. 1 +BLOCK CONTINUED

BLOCK 373(565B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780319   4 OF ** )
 1.                HERE SWAP !                      \ ; EXECUTION HERE
 2. DIRECT RPOP    RP ISZ,  NEXT,                   \ POP TOP OF
 3.                                                 \ RETURN STACK
 4. 1 +BLOCK CONTINUED
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 374(566B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780313   5 OF ** )
 1. <CODE ENTER  \ ENTER A WORD IN THE DICTIONARY
 2.                DP LDA,  LAST STA,              \ THIS IS LAST ENTRY
 3.                CURRENT ) LDB,                  \ SAVE HEAD ADDRESS
 4.                CURRENT ) STA,                  \ INCLUDE IN VOCABULARY
 5.           #CHF IFTRUE #WDS 1+ # ADA,           \ CHANGE DP VALUE BY WC
 6.                OTHERWISE A ) LDA,  AL8,        \ COUNT IN LOW BYTE
 7.                          377 # AND,  21 ( ARS ) , \ COUNT/2
 8.                          A0, NOT  INA,         \ COUNT >= 2
 9.                          2 # ADA,  DP ADA,     \ NEW DIC. POINTER
10.                IFEND
11.               DP STA,                          \ INCLUDE IN DICTIONARY
12.               -1 # ADA,                        \ ADDRESS OF LINK FIELD
13.               A ) STB,                         \ SET LINK FIELD
14.               2 # ADA,  A LDB,  SHOVE          \ COMPLETE 'CODE' ENTRY
15. 1 +BLOCK CONTINUED

BLOCK 375(567B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780319   6 OF ** )
 1. <CODE COLON  \ CORRECT ENTRY FOR ':' DEFINITION
 2.                0 ) # LDA,  STATE STA,          \ SET COMPILE MODE
 3.                CURRENT LDA,                    \ SEARCH IN THE VOC.
 4.                CONTEXT STA,                    \ WHERE DEFS ARE MADE
 5.                LAST LDB,                       \ ADDRESS OF LAST ENTRY
 6.           #CHF IFTRUE #WDS # ADB,              \ LINK ADDRESS OF ENTRY
 7.                OTHERWISE B ) LDA,  AL8,        \ COUNT IN LOW BYTE
 8.                          377 # AND,  21 ( ARS ) , \ COUNT/2
 9.                          A0, NOT  INA,         \ COUNT > 2
10.                          INA,  A ADB,          \ LINK ADDR. OF ENTRY
11.                IFEND
12.               B ) LDA,                         \ ADDRESS OF PREVIOUS D
13.               CURRENT ) STA,  NEXT,            \ EXCLUDE NEW ENTRY
14. 1 +BLOCK CONTINUED
15.

BLOCK 376(570B)
 0. ( METAFORTH BASIC VOCABULARY                    HN/780319   7 OF ** )
 1. HERE 10 #WDS 2 MAX + + ADR FORTH ! METAFORTH \ ':' EXECUTION
 2. :  :           RENAME $:                       \ COMPILE ':'
 3.                LIT  40 ,                       \ DELIMITER FOR WORD
 4.           0 OCT ,WORD                          \ FORWARD REFERENCE
 5.           0 OCT ,ENTER                         \ FORWARD REFERENCE
 6.                COLON                           \ CORRECT FOR ':'-DEF.
 7.                ;CODE                           \ APPEND FOLLOWING CODE
 8.                RP LDA, -1 # ADA, RP STA,       \ RESERVE LOCATION
 9.                INB,  RP ) STB,                 \ PUT NEW IC
10.                ' NEXT, 1+ JMP,
11. : ENTER RENAME $NTER ENTER RENAME ENTER ;
12. ' ENTER ,ENTER !       \ FIX FORWARD REFERENCE OF ENTER
13. \ ENTER IS REDEFINED TO ALLOW LOGS AND CHECKS
14. 4 +BLOCK ( 380D ) CONTINUED
15.
```

```
BLOCK 380(5740)
 0. ( METAFORTH LITERALS                                  HN/780319   1 OF ** )
 1.      0 OCT 'DLIT                                    \ DP-LIT ADDRESS
 2.
 3. CODE   ,       RENAME $,                            \ , SHOVES TOP OF STACK
 4. DIRECT S,      S) LDA,                              \ TAKE TOS
 5. DIRECT COMMA  DP ) STA,  DP ISZ,  POP,              \ SHOVE A, POP TOS
 6.
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 381(575B)
 0. ( METAFORTH LITERALS                                  HN/780319   2 OF ** )
 1. <CODE  LOC     \ PUSH OR COMPILE PARAMETER ADDRESS
 2.                \ OF ENTRY JUST FOUND
 3.                N 1+ LDA,                            \ ADDRESS OF ENTRY
 4.              #CHF IFTRUE #WDS 2 + # ADA,            \ PARAMETER FIELD
 5.                OTHERWISE A ) LDA, ALB,              \ COUNT IN LOW BYTE
 6.                           377 # AND,  21 ( ARS ) ,  \ COUNT/2
 7.                           A0, NOT  INA,             \ COUNT >= 2
 8.                           3 # ADA, N 1+ ADA,        \ PARAMETER FIELD
 9.                IFEND
10. DIRECT E16    STATE LDB,  B0, NOT                  \ IMMEDIATE MODE
11.               PUSH,                                \ YES, PUSH VALUE
12.               ' LIT # LDB,                         \ NO. SHOVE 'DEF LIT'
13. DIRECT C16A   DP ) STB,  DP ISZ,                   \ SHOVE B REGISTER
14.               A LDB,  SHOVE                        \ SHOVE VALUE
15. 1 +BLOCK CONTINUED

BLOCK 382(576B)
 0. ( METAFORTH LITERALS                                  JV/780403   3 OF ** )
 1. CODE    LITERAL  \ PUSH OR COMPILE NUMBER
 2.                  N 1+ LDA,  DPL LDB,
 3.                  B+,  E16                          \ 16-BIT LITERAL?
 4.                  STATE LDB, B0,                    \ DP LITERAL, STATE?
 5.                  IF,
 6. DIRECT  F32            SP LDB, -1 # ADB, SP STB, \ EX. DP LITERAL
 7.                        S) STA,                     \ PUSH L.S.P.
 8.                        N LDA,  PUSH,               \ PUSH M.S.P.
 9.                  THEN,                             \ COMPILE DP LITERAL
10.                  'DLIT LDB,  DP ) STB,
11.                  DP ISZ,                           \ SHOVE 'DEF DLIT'
12.                  N LDB,  C16A                      \ SHOVE DP NUMBER
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 383(577B)
 0. ( METAFORTH LITERALS                                JV/771007   4 OF ** )
 1. <CODE   DLIT  \ EXECUTE DP LITERAL
 2.               RP ) LDB,  B ) LDA,
 3.               N STA,                          \ STORE M.S.P.
 4.               RP ) ISZ,  INB,                 \ INCREMENT IC
 5.               B ) LDA,                        \ GET L.S.P.
 6.               RP ) ISZ,                       \ INCREMENT IC
 7.               E32                             \ PUSH DP NUMBER
 8.
 9. ' DLIT 'DLIT !                  \ SET FORWARD REFFRENCE
10.
11. 7 +BLOCK ( 390D ) CONTINUED
12.
13.
14.
15.

BLOCK 390(606B)
 0. ( METAFORTH LEVEL 0 WORDS                           JV/780320   1 OF ** )
 1. CODE 2DROP   ' 2POP, ADDRESS !                          RENAME $DROP
 2. CODE DROP    '  POP, ADDRESS !                          RENAME $ROP
 3. CODE @       S) LDA,  A ) LDA,  PUT,                    RENAME $@
 4. CODE !       S1) LDA,  S) LDB,  B ) STA, 2POP,          RENAME $!
 5. CODE +       S) LDA,  S1) ADA,  BINARY,                 RENAME $+
 6. CODE 1+      S) ISZ, NEXT,  NEXT,                       RENAME $1+
 7. CODE SWAP    S1) LDA,  S) LDB,  S1) STB,  PUT,          RENAME $WAP
 8. CODE OVER    S1) LDA,  PUSH,                            RENAME $VER
 9. CODE DUP     S) LDA,  PUSH,                             RENAME $UP
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 391(607B)
 0. ( METAFORTH LEVEL 0 WORDS                           HN/780319   2 OF ** )
 1. : CONSTANT                          RENAME $ONSTANT
 2.              LIT  40 ,
 3.           0 OCT ..WORD                           \ WORD
 4.              ENTER  $,  ;CODE
 5.    HERE      1CON FORTH ! METAFORTH
 6.              INB,  B ) LDA,  PUSH,
 7.
 8. : VARIABLE                          RENAME $ARIABLE
 9.              $ONSTANT                           \ CONSTANT
10.              ;CODE
11.    HERE      1VAR FORTH ! METAFORTH
12.              INB,  B LDA,  PUSH,
13.
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 392(610B)
 0. ( METAFORTH LEVEL 0 WORDS                          HN/780319   3 OF ** )
 1. CODE  COUNT  S) LDB,                              \ STRING ADDRESS
 2.              B ) LDA,  AL8,                       \ COUNT IN LOW BYTE
 3.              377 # AND,                           \ CHARACTER COUNT
 4.              RBL,  INB,  S) STB,                  \ BYTE ADDR. ON STACK
 5.              PUSH,                                \ PUSH CHARACTER COUNT
 6.
 7. CODE NULL    ' NEXT, ADDRESS !
 8.
 9. 15 +BLOCK ( 405D ) CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 405(625B)
 0. ( METAFORTH RESOURCE SHARING                       HN/780319   1 OF ** )
 1. CODE ALLOCATE \ ALLOCATE A DEVICE TO A USER
 2.               \ DEVICE-USER ADDRESS ON STACK
 3.               HERE S) LDA, A ) LDB,               \ CURRENT USER
 4.                    U BNE,                         \ IS IT YOU?
 5.                    IF, SWAP                       \ NO, IT IS NOT YOU
 6.                        B0, NOT                    \ IS IT USED?
 7.                        IF, SWAP                   \ YES, BY SOMEONE ELSE
 8.                            PAUSE,                 \ WAIT UNTIL RELEASED
 9.               JMP,                                \ BY CHECKING AGAIN
10.                    THEN,
11.                    U LDB,                         \ EXECUTING USER ( YOU )
12.                    A ) STB,                       \ IS CURRENT USER
13.                THEN,
14.               POP,
15. 1 +BLOCK CONTINUED

BLOCK 406(626B)
 0. ( METAFORTH RESOURCE SHARING                       JV/780321   2 OF ** )
 1. CODE RELEASE  \ RELEASE A DEVICE FROM A USER
 2.               \ DEVICE-USER ADDRESS ON STACK
 3.               S) LDA, A ) LDB,                    \ CURRENT USER
 4.               U BNE,                              \ IS IT YOU?
 5.               RSS,                                \ NO, ILLEGAL RELEASE
 6.               IF,  CLB,  A ) STB,                 \ YES, RELEASE DEVICE
 7.               THEN,
 8.               POP,
 9.
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 407(627B)
 0. ( METAFORTH RESOURCE SHARING  DMA          JV/780320   3 OF ** )
 1. COMPUTER 000001 AND                  \ DMA AVAILABLE ?
 2.      IFTRUE
 3. VAR DMA      RENAME $MA
 4.              0 OCT DMA               \ DMA USER ADDRESS
 5.              0 ,                     \ TEMPORARY A-REGISTER
 6.              0 ,                     \ INTERRUPT ENTRY POINT
 7.              DMA 1+ STA,             \ SAVE A-REGISTER
 8.              DMA ) LDA,              \ GET USER STATUS
 9.              137777 # AND,           \ CLEAR DMA-WAIT OF USER
10.              DMA ) STA,              \ RESTORE USER STATUS
11.              DMA 1+ LDA,             \ RESTORE A-REGISTER
12.              6 CLC,
13.              DMA 2 + ) JMP,          \ RETURN TO PROGRAM
14.      IFEND
15. 3 +BLOCK ( 410D ) CONTINUED

BLOCK 410(632B)
 0. ( METAFORTH CHARACTER HANDLING            JV/780330   1 OF  2 )
 1. CR ." CHARACTER HANDLING " HERE .
 2. SUBROUTINE FETCH     \ FETCH CHARACTER FROM STRING
 3.                      IP LDA,  CAER,  \ BYTE ADDR. => ADDR. POSIT.
 4.                      A ) LDA,        \ FETCH WORD
 5.                      E0, NOT  ALS,   \ ROTATE WHEN HIGH BYTE
 6.                      377 # AND,      \ CHARACTER!
 7.                      IP ISZ,         \ INCREMENT BYTE POINTER
 8.                      RETURN
 9.
10. CODE B@  S) LDA, IP STA, FETCH PUT,
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 411(633B)
 0. ( METAFORTH CHARACTER HANDLING            JV/780330   2 OF  2 )
 1. SUBROUTINE DEPOSIT \ DEPOSIT A CHARACTER IN A STRING
 2.                      OP LDB,  CBER,  \ BYTE ADDR. => ADDR. POSIT.
 3.                      CH STB,  OP ISZ, \ SAVE ADDR, INCR. POINTER
 4.                      E0, NOT  ALS,   \ ROTATE WHEN HIGH BYTE
 5.                      A LDB,          \ SAVE CHARACTER
 6.                      377 # LDA,      \ GET MASK
 7.                      E0,  ALS,       \ SHIFT MASK?
 8.                      CH ) AND,       \ DESTROY BYTE
 9.                      B IOR,          \ MERGE CHARACTERS
10.                      CH ) STA,       \ STORE WORD
11.                      RETURN
12.
13. CODE B!  S) LDA, OP STA, S1) LDA, DEPOSIT 2POP,
14.
15. 4 +BLOCK ( 415D ) CONTINUED
```

```
BLOCK 415(6378)
 0. ( METAFORTH WORD HANDLING                          HN/770928   1 OF ** )
 1. CR ." WORD HANDLING " HERE .
 2. <CODE WORD1  S) LDB,  RBL,                    \ INPUT BYTE ADDRESS
 3.              IN LDA,  IN STB,                 \ SAVE START ADDRESS
 4.              IN ADA,  IP STA,                 \ SET INPUT BYTE POINTER
 5.              DP LDB,  CLA,  B ) STA,          \ ZERO TOP OF DICTIONARY
 6.              INB,  3 ) STA,                   \ ZERO SECOND WORD
 7.              RBL,  -1 # ADB,  DP STB,         \ OUTPUT BYTE POINTER
 8. #CHF 3 > IFTRUE  \ ZERO MORE WORDS WHEN FIXED #CHF > 3
 9.                #WDS 2 - MINUS # LDB,  N STB, \ # TO BE ZEROED
10.                   DP LDB,  2 # ADB,          \ FIRST WORD
11.                   HERE  B ) STA,  INB,       \ ZERO WORD
12.                         N ISZ,               \ ALL DONE?
13.                   JMP,                       \ DO ONE MORE
14.          IFEND
15. 1 +BLOCK CONTINUED

BLOCK 416(640B)
 0. ( METAFORTH WORD HANDLING                          HN/780319   2 OF ** )
 1.              HERE  FETCH  S1) ANE, JMP, \ SKIP DELIMITER FIRST
 2.              HERE  DEPOSIT              \ DEPOSIT CHAR
 3.                    400 # LDA,  DP ) ADM,   \ UPDATE CHAR. COUNT
 4.                    TCA,  41000 # ADA,  A+, \ LENGTH < MAX ( 64D
 5.                    IF,  SWAP              \ LENGTH O.K.
 6.                         FETCH S1) ANE,    \ NEXT CHAR DELIM?
 7.                         RSS,              \ SKIP JUMP
 8.              JMP,                         \ NO, HANDLE NEXT CHAR
 9.                    THEN,                  \ YES, FINISH WORD
10.              40 # LDA,  DEPOSIT           \ DEPOSIT BLANK
11.              IN LDA,  TCA,                \ -START OF INPUT BUFFER
12.              IP ADA,  IN STA,             \ INTERPRETER POINTER
13.              2POP,                        \ ADDRESS AND DELIMITER
14. 1 +BLOCK CONTINUED
15.

BLOCK 417(641B)
 0. ( METAFORTH WORD HANDLING                          JV/780403   3 OF ** )
 1. <CODE SCREEN  SCR LDA,  A0,               \ MASS STORAGE OR TTY?
 2.               PUSH,                       \ BLOCK NUMBER
 3.               PUSH=SKIP                   \ SKIP NEXT FORTH INSTR.
 4. 3 +BLOCK ( 420D ) CONTINUED
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 420(644B)
 0. ( METAFORTH NUMBER CONVERSION                    JV/780315   1 OF ** )
 1. CODE  NUMBER DP LDA, 22 ( RAL ) , INA, \ BYTE ADDRESS
 2.              IP STA,
 3.              CLA,  N STA,  N 1+ STA,  \ ZERO NUMBER
 4.              0 ) # XOR,  DPL STA,     \ NO DECIMAL POINT YET
 5.              FETCH  -55 # ADA,  SIGN STA, \ SIGN= CHAR= '-'
 6.              A0, NOT                  \ NEGATIVE
 7.              IF,  IP LDA, -1 # ADA, IP STA, \ POSITIVE, RESET IP
 8.              THEN,
 9.              HERE BASE LDA,  BASE0 STA, \ SET CONVERSION BASE
10.                   FETCH  CH STA,      \ FETCH CHARACTER
11.                   -60 # ADA,  A+,     \ CHAR >= '0' ?
12.                   IF,  OVER           \ YES
13.                        BASE0 SUB, A-, \ DIGIT < BASE0?
14.                        IF, SWAP       \ YES, VALID DIGIT
15. 1 +BLOCK CONTINUED

BLOCK 421(645B)
 0. ( METAFORTH NUMBER CONVERSION                    JV/780118   2 OF ** )
 1.                             BASE0 ADA,  CH STA, \ DIGIT VALUE
 2.                             N LDA,              \ M.S.P.
 3.                             BASE0 MPY,  N STA,  \    * BASE0
 4.                             N 1+ LDA,           \ L.S.P.
 5.                             BASE0 MPY,          \    * BASE0
 6.                             B+,  BASE0 ADB,     \ NEG L.S.P.?
 7.                             CLE,  CH ADA,       \ ADD DIGIT
 8.                             E0,  INB,           \ ADD CARRY
 9.                             N ADB,              \ ADD M.S.P.
10.                             N STB,  N 1+ STA,   \ STORE NUMBER
11.                             DPL ISZ,            \ INCREMENT DPL
12.                   JMP,                          \ NEXT DIGIT
13.                        THEN,
14.                   THEN,
15. 1 +BLOCK CONTINUED

BLOCK 422(646B)
 0. ( METAFORTH NUMBER CONVERSION                    HN/780319   3 OF ** )
 1.                        CH LDA,  56 # ANE,     \ NO DECIMAL POINT?
 2.                        IF,      72 # ANE,     \ NO SPECIAL CHAR?
 3.                             IF,  40 # ANE,    \ NO BLANK?
 4.                                  IF,          \ NO VALID NUMBER
 5. DIRECT 2SKIP                          RP ) ISZ, \ SKIP 2 INSTR.
 6. DIRECT 1SKIP                          RP ) ISZ, \ SKIP 1 INSTR.
 7.                                       NEXT,
 8.                                  THEN,        \ BLANK, VALID NUMBER.
 9.                                  SIGN LDA,    \ GET SIGN
10.                                  A0,  NEXT,   \ POSITIVE
11.                                  N LDB,  N 1+ LDA, \ NEGATIVE NUMBER
12.                                  CMB,  3007 ( CMA,INA,SZA,RSS ) ,
13.                                  INB,         \ COMPLEMENT NUMBER
14.                                  N STB,  N 1+ STA, \ STORE NUMBER
15. 1 +BLOCK CONTINUED
```

```
BLOCK 423(6478)
 0. ( METAFORTH NUMBER CONVERSION                    HN/780319   4 OF ** )
 1.                                NEXT,
 2.                             THEN,                \ SEXIGESIMAL NUMBER
 3.                              6 # LDA,
 4.                    OVER 1+ JMP,                  \ NEXT DIGIT BASE 6
 5.                          THEN,                   \ DECIMAL POINT
 6.                          CLA,  DPL STA,          \ ZERO DPL
 7.                    JMP,                          \ CONVERT NEXT DIGIT
 8. 7 +BLOCK ( 430D ) CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 430(656B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780313   1 OF  7 )
 1. CR  ." DICTIONARY SEARCH "  HERE .
 2. CODE  FIND        \ FIND A DICTIONARY ENTRY
 3.
 4.       #CHF  3  =  IFTRUE      1 +BLOCK  LOAD   IFEND
 5.       #CHF  3  >  IFTRUE      3 +BLOCK  LOAD   IFEND
 6.       #CHF  0  =  IFTRUE      5 +BLOCK  LOAD   IFEND
 7.
 8. 12  +BLOCK  ( 440D )  CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 431(657B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780313   2 OF  7 )
 1.             DP LDA,  INA,  A ) LDA,  N STA,  \ N := (DP+1)
 2.             DP ) LDB,                        \ FOR QUICK LOOP
 3.             CONTEXT ) LDA,                   \ SEARCH STARTS HERE
 4.             HERE A ) BNE,                    \ FIRST WORD EQUAL?
 5.                  IF, HERE #WDS # ADA,        \ NO, GET LINK ADDR.
 6.                          A ) LDA,            \ ADDR. OF NEXT ENTRY
 7.                          A0,                 \ END OF VOCABULARY?
 8.       ROT  JMP,                              \ NO, TRY NEXT ENTRY
 9.                          2SKIP               \ YES, WORD NOT FOUND
10.       SWAP       THEN,                       \ FIRST WORD EQUAL
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.
```

```
BLOCK 432(660B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780319   3 OF  7 )
 1.                          N 1+ STA,               \ SAVE ENTRY ADDRESS
 2.                          INA,                    \ CHAR 2&3
 3.                          A ) LDA,                \ GET THEM
 4.                          77577 # AND,            \ MASK OUT BITS
 5.                          N ANE,                  \ CHAR 2&3 EQUAL?
 6.                          NEXT,                   \ FOUND !!!!!
 7.                          N 1+ LDA,               \ GET ADDRESS
 8.                    JMP,                          \ AND TRY NEXT ENTRY
 9. ;S
10.
11.
12.
13.
14.
15.

BLOCK 433(661B)
 0. ( METAFORTH DICTIONARY SEARCH                    JV/780316   4 OF  7 )
 1.               DP ) LDA,                          \ FOR QUICK LOOP
 2.               CONTEXT ) LDB,                     \ SEARCH STARTS HERE
 3.               HERE B ) ANE,                      \ FIRST WORD EQUAL?
 4.                    IF, HERE #WDS # ADB,          \ NO, GET LINK ADDR.
 5.                          B ) LDB,                \ ADDR. OF NEXT ENTRY
 6.                          B0,                     \ END OF VOCABULARY?
 7.       ROT  JMP,                                  \ NO, TRY NEXT ENTRY
 8.                          2SKIP                   \ YES, WORD NOT FOUND
 9.       SWAP       THEN,                           \ FIRST WORD EQUAL
10.                          N 1+ STB,               \ SAVE ENTRY ADDRESS
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 434(662B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780319   5 OF  7 )
 1.                    #WDS 1- MINUS # LDA,          \ INITIALIZE
 2.                          N STA,                  \    WORD-COUNT
 3.                          DP LDA,                 \ INITIALIZE
 4.                          N 2 + STA,              \   LOOP ADDRESS
 5.                          HERE N 2 + ISZ,         \ INCREMENT
 6.                               INB,               \   LOOP ADDRESS
 7.                               B ) LDA,           \ GET WORD
 8.                               77577 # AND,       \ MASK OUT PRECEDENCE
 9.                               N 2 + ) ANE,       \ EQUAL ?
10.                               IF, N 1+ LDB,      \ NO, RESTORE
11.                                   DP ) LDA,
12. ROT                     JMP,      THEN,          \ CONTINUE SEARCH
13.                               N ISZ,             \ ALL COMPARED ?
14.                          JMP,                    \ NO, AGAIN
15.                          NEXT,     ;S            \ YES, FOUND!
```

```
BLOCK 435(663B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780208   6 OF  7 )
 1.              DP ) LDA, ALB,
 2.              377 # AND,  21 ( ARS ) ,
 3.              A0, NOT  INA,
 4.              TCA,  N 3 + STA,                    \ INITIALIZE WORD+COUNT
 5.              DP ) LDA,  N 4 + STA,               \ SAVE INDIRECT ADDRESS
 6.              CONTEXT ) LDB,                      \ SEARCH STARTS HERE
 7.              HERE B ) LDA,                       \ GET FIRST WORD
 8.                   N 4 + ANE,                     \ FIRST WORD EQUAL?
 9.                   IF, HERE ALB,
10.                            377 # AND, 21 ( ARS ) ,
11.                            A0, NOT INA,
12.                            INA,  A ADB,          \ LINK ADDRESS
13.                            B ) LDB,              \ ADDRESS OF NEXT ENTRY
14.                            B0,                   \ END OF VOCABULARY?
15.         ROT JMP,     1 +BLOCK CONTINUED

BLOCK 436(664B)
 0. ( METAFORTH DICTIONARY SEARCH                    HN/780319   7 OF  7 )
 1.                           2SKIP                 \ YES, NOT FOUND
 2.         SWAP      THEN,                         \ FIRST WORD EQUAL
 3.                           N 1+ STB,             \ SAVE ADDRESS
 4.                           N 3 + LDA, N STA,     \ INITIALIZE WRD-CNT
 5.                           DP LDA, N 2 + STA,    \ AND LOOP ADDRESS
 6.                           HERE N 2 + ISZ,
 7.                                INB,
 8.                                B ) LDA,
 9.                                77577 # AND,
10.                                N 2 + ) ANE,
11.                                IF, N 1+ LDB,
12.                                    B ) LDA,
13.         ROT JMP,               THEN,
14.                                N ISZ,
15.                           JMP,  NEXT,     ;S

BLOCK 440(670B)
 0. ( METAFORTH BLOCK HANDLING                       JV/780201   1 OF **  )
 1. CR ." BLOCK HANDLING " HERE .
 2. : BLOCK-MAP    $ROP  ;                           \ MAP A DEVICE
 3. : BLOCK-READ   $DROP ;                           \ READ A BLOCK
 4. : BLOCK-WRITE $DROP ;                            \ WRITE A BLOCK
 5. : GET    LIT OWNER ,  ALLOCATE ;                 \ ALLOCATE BLOCKS
 6. : TEG    LIT OWNER ,  RELEASE  ;                 \ RELEASE  BLOCKS
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 441(671B)
 0. ( METAFORTH BLOCK HANDLING                          JV/780313   2 OF ** )
 1. MEMORY CONSTANT HIGH                  \ HIGH LIMIT OF BUFFER AREA
 2. MEMORY BLOCK-LENGTH 3 + #-BUFFERS * - SET MEMORY
 3. MEMORY CONSTANT LOW                   \ LOW  LIMIT OF BUFFER AREA
 4.
 5. VAR PREV                          RENAME $REV
 6. LOW                          OCT PREV \ ADDRESS OF PREV BUFFER
 7. LOW BLOCK-LENGTH 3 + + OCT ALT  \ ADDRESS OF ALT BUFFER
 8.
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 442(672B)
 0. ( METAFORTH BLOCK HANDLING                          JV/780330   3 OF ** )
 1. <: BREAD   $REV  $@  $1+  $VER          \ PREV @ 1+ OVER
 2.            BLOCK-READ                   \ BUFFERADDR, BLOCK#
 3.            $REV  $@  $WAP  $VER         \ PREV @ SWAP OVER
 4.            $!  $1+                      \ ! 1+
 5.            $UP  LIT 1000 , $+           \ DUP 1000 +
 6.            LIT  20076 ,  $VER  $!       \ ' >' OVER !
 7.            LIT  41040 ,  $WAP  $1+      \ 'B ' SWAP 1+
 8.            $!  ;                        \ ! ;
 9. <: BWRITE  $REV  $@  $1+                \ PREV @ 1+
10.            $REV  $@  $@                 \ PREV @ @
11.            BLOCK-WRITE  ;               \ BUFFERADDR, BLOCK#
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 443(673B)
 0. ( METAFORTH BLOCK HANDLING                          HN/780319   4 OF ** )
 1. <CODE ;BUFFER HERE ALT LDA,  A LDB,     \ ALTERNATE BUFFER
 2.                    BLOCK-LENGTH 3 + # ADA,
 3.                    HIGH # ANE,          \ NEW ALT BUFFER = HIGH?
 4.                    LOW  # LDA,          \ YES, ALT:=LOW
 5.                    ALT STA,             \ NEW ALT BUFFER
 6.                    PREV BNE,            \ PREV BUF = OLD ALT BUF?
 7.                JMP,                     \ YES, REPEAT ALLOCATION
 8.                PREV STB,                \ THIS IS THE NEW PREV
 9.                B ) LDA,  A+,            \ BUFFER MARKED UPDATED?
10.                IF,  RP ISZ,  NEXT,      \ NO, POP IC
11.                THEN,                    \ YES
12.                77777 # AND,  B ) STA,   \ CLEAR UPDATE BIT
13.                NEXT,
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 444(674B)
 0. ( METAFORTH BLOCK HANDLING                          HN/780319   5 OF ** )
 1. <CODE *CORE PREV ) LDA,  77777 # AND, \ BK# OF PREV BUFFER
 2.                 S) ANE,               \ FOUND?
 3.                 IF, HIGH # LDB,       \ NO, SEARCH ALL BUFFERS
 4.                     HERE BLOCK-LENGTH 3 + MINUS # ADB, \ NEXT BUF
 5.                          B ) LDA,     \ BLOCK #
 6.                          77777 # AND, \ MASK OFF UPDATE FLAG
 7.                          S) ANE,      \ FOUND?
 8.                          IF, SWAP     \ NOT FOUND
 9.                              LOW # BNE, \ ANY MORE BUFFERS?
10.                              NEXT,    \ NO !!!!
11.                     JMP,              \ YES, TRY NEXT BUFFER
12.                          THEN,        \ FOUND IN ANY BUFFER
13.                          PREV STB,    \ SET THIS ONE TO PREV
14.                 THEN,                 \ FOUND IN PREV BUFFER
15. 1 +BLOCK CONTINUED

BLOCK 445(675B)
 0. ( METAFORTH BLOCK HANDLING                          HN/780319   6 OF ** )
 1.                 RP ISZ,               \ POP IC
 2.                 PREV LDA,  INA,  PUT, \ PUT BUFFER ADDRESS
 3. 1 +BLOCK CONTINUED
 4.
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 446(676B)
 0. ( METAFORTH BLOCK HANDLING                          JV/780330   7 OF ** )
 1. CODE ERASE-CORE \ MARK ALL BUFFERS EMPTY
 2.                 HIGH # LDB,  CLA,
 3.                 HERE BLOCK-LENGTH 3 + MINUS # ADB,
 4.                      B ) STA,         \ MARK EMPTY
 5.                      LOW # BNE,       \ END OF BUFFER AREA?
 6.                      NEXT,            \ YES, EXIT
 7.                 JMP,                  \ NO, DO NEXT BUFFER
 8.
 9. CODE UPDATE     \ FLAG PREV BUFFER AS UPDATED
10.                 PREV ) LDA,  0 ) # IOR,
11.                 PREV ) STA,  NEXT,
12.
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 447(677B)
 0. ( METAFORTH BLOCK HANDLING                    JV/780330   8 OF ** )
 1. <: BUFFER    ;BUFFER  BWRITE ;
 2.  : ABLOCK    GET      *CORE    BUFFER  BREAD  ;
 3.  : BLOCK     NULL     ABLOCK  ;
 4. <: $BLOCK    *CORE    BUFFER
 5.               $REV  $@  $WAP  $VER       \ PREV @ SWAP OVER
 6.               $!  $1+  ;                 \ ! 1+ ;
 7.  : BUFFER    NULL     GET  $BLOCK   UPDATE  ;  RENAME $UFFER
 8.
 9. : READ-MAP  GET BUFFER  $REV $@ $1+ $UP
10.             BLOCK-MAP  LIT  @ ,  $WAP  $!  ;
11. : FLUSH  \ #-BUFFERS @ DO BUFFER LOOP ;
12.          GET
13.          #-BUFFERS    1 +BLOCK LOAD    DROP
14.          ;
15. 2 +BLOCK CONTINUED

BLOCK 448(700B)
 0. ( METAFORTH BLOCK HANDLING                    JV/770929   9 OF ** )
 1. BUFFER                                  \ ASSEMBLE BUFFER
 2. 1-                                      \ DECREMENT COUNT
 3. DUP IFTRUE @ +BLOCK CONTINUED           \ DO REST
 4.     IFEND
 5. ;S
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 449(701B)
 0. ( METAFORTH BLOCK HANDLING                    HN/780319  10 OF ** )
 1. CODE ?U    S) LDA,
 2.            HIGH MINUS # ADA,
 3.            A-,  NEXT,
 4.            HIGH LOW - # ADA,
 5.            A+,  NEXT,
 6.            CLB,
 7.            BLOCK-LENGTH 3 + # DIV,
 8.            BLOCK-LENGTH 3 + # MPY,
 9.            LOW # ADA,
10.            A LDB,
11.            @ ) # LDA,  B ) IOR,  B ) STA,
12.            NEXT,
13. 13 +BLOCK ( 460D ) CONTINUED
14.
15.
```

```
BLOCK 460(714B)
 0. ( METAFORTH BLOCK LOADING                          HN/780319   1 OF ** )
 1. CR ." BLOCK LOADING "  HERE .
 2. CODE  LOAD                                    RENAME $OAD
 3.                RP LDA,              -1 # ADA, \ SAVE IC
 4.                IN LDB,  A ) STB,    -1 # ADA, \ SAVE IN
 5.               SCR LDB,  A ) STB,    -1 # ADA, \ SAVE SCR
 6.                RP STA,                       \ UPDATE RP
 7. DIRECT CCC  CLB,  IN STB,                    \ RESET IN
 8.                S) LDB,  SCR STB,             \ SET BLOCK #
 9.                GO LDB,  RP ) STB,            \ SET IC !!!!
10.                POP,
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 461(715B)
 0. ( METAFORTH BLOCK LOADING                          JV/780124   2 OF ** )
 1.
 2. CODE  ;S            RENAME $S
 3.                     RP ISZ,                 \ DESTROY IC
 4.                     RP ) LDB,  SCR STB,     \ RESTORE SCR
 5.                     RP ISZ,
 6.                     RP ) LDB,  IN STB,      \ RESTORE IN
 7.                     RPOP                    \ POP SCR
 8.
 9. CODE  CONTINUED  RENAME $ONTINUED
10.                     ALT LDA,  PREV LDB,
11.                     ALT STB,  PREV STA,
12.                     CCC
13.
14. 1 +BLOCK CONTINUED
15.

BLOCK 462(716B)
 0. ( METAFORTH BLOCK LOADING                          HN/780319   3 OF ** )
 1.
 2. CODE >B    IMMEDIATE
 3.            DSP,  SCR LDA,  INA,  S) STA,
 4.            ' $ONTINUED 1+ JMP,
 5.
 6. CODE  +BLOCK       RENAME $BLOCK
 7.                    SCR LDA,  S) ADA,  PUT,
 8.
 9. 10 +BLOCK ( 470D ) CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 470(726B)
 0. ( METAFORTH INTERPRETER INPUT                    JV/771006   1 OF ** )
 1. CR  ." METAFORTH INTERPRETER INPUT " HERE .
 2.
 3. : WORD    HERE 1-  .WORD !          \ FIX WORD IN :
 4.           HERE 1- ..WORD !          \ FIX WORD IN CONSTANT
 5.                SCREEN               \ MASS-STORAGE OR CONSOLE
 6.                BLOCK                \ MASS-STORAGE GET BLOCK
 7.                WORD1                \ READ WORD
 8.                ;
 9.
10. 5 +BLOCK ( 475D ) CONTINUED
11.
12.
13.
14.
15.

BLOCK 475(733B)
 0. ( METAFORTH FORTH VOCABULARY                     HN/780313   1 OF ** )
 1. CR ." FORTH VOCABULARY " HERE .
 2. #WDS 2 MAX 3 + ,CODE FORTH IMMEDIATE      RENAME $ORTH
 3.                1600 ,                     \ PSEUDO ENTRY HEAD
 4.                1600 ,                     \ PSEUDO ENTRY LINK
 5.          #WDS 2 - 0 MAX  DP+!             \ COMPLETE ENTRY
 6.                   0 ,                     \ VOCABULARY HEAD
 7.                   0 ,                     \ VOCABULARY LINK
 8.  ' ASSEMBLER 2 + #WDS 2 MAX + ,           \ VOCABULARY CHAIN
 9.  HERE 2 -  VOC-LINK USER-AREA + !         \ SET VOCABULARY CHAIN
10.                INB,                       \ VOC. HEAD ENTRY ADDR.
11.  ' ASSEMBLER 2 + #WDS 2 MAX + STB,        \ SET ASSEMBLER LINK
12.              -1 # ADB,                    \ RESET B-REGISTER
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 476(734B)
 0. ( METAFORTH FORTH VOCABULARY                     HN/780319   2 OF ** )
 1.  HERE ' ASSEMBLER !                       \ ASSEMBLER JUMPS HERE
 2.                #WDS 2 MAX 1+ # ADB,       \ ADDRESS OF HEAD
 3.                CONTEXT STB,               \ SET CONTEXT *
 4.                NEXT,
 5. 4 +BLOCK ( 480D ) CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 480(740B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                HN/780313   1 OF ** )
 1. CR ." ASSEMBLER VOCABULARY " HERE .
 2. ASSEMBLER-DEFINITIONS
 3. 0 OCT        #?
 4. SUBROUTINE  #                                RENAME ##
 5.                    S) LDB,                   \ GET VALUE
 6.                    #1 @ # LDA,               \ FIRST ADDRESS
 7.                    HERE
 8.                          A ) BNE,            \ FOUND?
 9.             OVER         RETURN              \ YES, RETURN
10.                          INA,                \ NO, INCREMENT ADDR.
11.                          #? ANE,             \ ANY MORE?
12.                          RSS,                \ NO, SKIP NEXT INSTR.
13.                    JMP,                      \ TEST NEXT CONSTANT
14. 1 +BLOCK CONTINUED
15.

BLOCK 481(741B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                HN/780319   2 OF ** )
 1.                    #1 @ #-LENGTH + # ANE,    \ AREA FULL ?
 2.                    IF,                       \ YES, ABORT
 3.                          A ) STB,            \ NO, STORE IN AREA
 4.                             #? ISZ,
 5.             SWAP         RETURN
 6.                    THEN,
 7.                    3 # LDA,                  \ ERROR NUMBER
 8.                    ERROR,                    \ ISSUE ERROR MESSAGE
 9.
10. CODE  #                                     RENAME ###
11.                    ##                        \ REPLACE BY ADDRESS
12.                    PUT,                      \ VALUE ON STACK
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 482(742B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                JV/780320   3 OF ** )
 1. <CODE  STH     S) LDA,  ALS,
 2.   HERE                                        \ FOR AFIOU
 3.                ALS,  PUT,
 4. <CODE  AFIOU   S) LDA,  1323 ( RAR,RAR ) ,
 5.                200 # IOR,
 6.                JMP,
 7.
 8. :  MR  STH $ONSTANT ;CODE                     \ MEMORY REF. INSTR.
 9.        S) LDA,                                \ OPERAND ADDRESS
10.        101777 # AND,
11.        S) ANE,                                \ IS IT ON BASE PAGE?
12.
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 483(743B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                JV/780117   4 OF ** )
 1.       IF,  S) LDA,                              \ NO, NOT ON BASE PAGE!
 2.            DP XOR,
 3.            76000 # AND,
 4.            A0,                                  \ CURRENT PAGE?
 5.            IF,  S) LDA,                         \ YES, STILL POSSIBLE
 6.                 101777 # AND,                   \ MASK OFF PAGE BIT
 7.                    2000 # IOR,                  \ SET CURRENT PAGE BIT
 8.            ELSE,                    \ NOT ON BASE OR CURRENT PAGE
 9.                 N 1+ STB,                       \ SAVE B-REGISTER
10.                 ##                              \ LINK ADDRESS
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 484(744B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                HN/780319   5 OF ** )
 1.                 N 1+ LDB,                       \ RESTORE B-REGISTER
 2.                 0 ) # IOR,                      \ INCLUDE INDIRECT BIT
 3.            THEN,
 4.       THEN,
 5. DIRECT SL1 INB,  B ) ADA,                       \ ADD INSTRUCTION CODE
 6.            COMMA                               \ COMPILE INSTRUCTION
 7.
 8. :  CPU     $ONSTANT ;CODE
 9.  HERE  1CPU  FORTH ! METAFORTH                  \ FOR 2CPU ASSEMBLY
10.            INB,  B ) LDA,                      \ GET INSTRUCTION CODE
11.            DP ) STA,  DP ISZ,                  \ INCLUDE IN DICT.
12.            NEXT,
13.
14. 1 +BLOCK CONTINUED
15.

BLOCK 485(745B)
 0. ( METAFORTH ASSEMBLER VOCABULARY                HN/780319   6 OF ** )
 1. :  DEAU    AEIOU $ONSTANT ;CODE                 \ GET INSTR. CODE
 2.            INB,  B ) LDA,                      \ GET INSTRUCTION CODE
 3.            DP ) STA,  DP ISZ,                  \ COMPILE INSTR. CODE
 4.            S,                                  \ COMPILE OPERAND ADDR.
 5.
 6. :  I/O     AEIOU $ONSTANT ;CODE
 7.            S) LDA,  77 # AND,                  \ GET SELECT CODE
 8.            SL1
 9.
10. CODE  )    RENAME $)
11.            S) LDA,  0 ) # IOR,  PUT,           \ SET INDIRECT BIT
12.
13. CODE  EDOC  CURRENT LDA,  CONTEXT STA,  NEXT,
14.
15. 1 +BLOCK CONTINUED
```

```
BLOCK 486(746B)
 0. ( METAFORTH ASSEMBLER VOCABULARY          JV/780320   7 OF ** )
 1.
 2. 2CPU NEXT,    RENAME $EXT,   NEXT,
 3.
 4. 2CPU 2POP,    RENAME $POP,   2POP,
 5.
 6. 2CPU ABORT,   RENAME $BORT,  ABORT,
 7.
 8. 2CPU ERROR,   RENAME $RROR,  ERROR,
 9.
10. 2CPU PAUSE,   RENAME $AUSE,  PAUSE,
11.
12. FORTH-DEFINITIONS
13.
14. 11 +BLOCK ( 495D ) CONTINUED
15.

BLOCK 495(757B)
 0. ( METAFORTH GENERAL CONSOLE I/O           JV/780331   1 OF ** )
 1. CR ." GENERAL CONSOLE I/O " HERE .
 2.
 3. CODE  TYPE    TALK # LDB,  U ADB,  B ) LDB,  \ GET ADDRESS OF
 4.                                    \ TYPE ROUTINE ( INDEXED )
 5.               S) LDA,              \ GET CHARACTER COUNT
 6.               A0,                  \ COUNT = 0 ?
 7.               .NEXT,               \ NO, TYPE !!
 8.               2POP,                \ YES, IGNORE
 9.
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 496(760B)
 0. ( METAFORTH GENERAL CONSOLE I/O           JV/780331   2 OF ** )
 1.
 2. CODE  EXPECT  DSP,                 \ RESERVE ONE LOCATION
 3.               S1) LDA,  S) STA,    \ CHARACTER COUNT ON TOP
 4.               MSG LDA,  A ADA,     \ MSG BYTE ADDRESS
 5.               S1) STA,             \ BYTE ADDRESS BELOW
 6.               ASK # LDB,  U ADB,   \ ADDRESS OF INPUT ROUTINE
 7.               B ) LDB,  .NEXT,     \ GO INPUT CHARACTERS
 8.
 9. : QUIT        TEG  LIT 120 , EXPECT ;
10.
11. 4 +BLOCK ( 500D ) CONTINUED
12.
13.
14.
15.
```

```
BLOCK 500(764B)
 0. ( METAFORTH INTERPPETER WORDS                    JV/780330   1 OF ** )
 1. CR ." INTERPRETER WORDS " HERE .
 2. <CODE  Q1     \ SET IC TO GO SEQUENCE
 3.              GO LDA,  RP ) STA,  NEXT,
 4.
 5.  HERE COMPILE ?+   477 ,                  \ " ?"
 6.                    TEG
 7.                    COUNT TYPE             \ ERROR MESSAGE
 8.                    LIT ?+ COUNT TYPE      \ TYPE QUESTION MARK
 9.                    LIT 120 , EXPECT       \ GET NEW INPUT LINE
10.                    Q1                     \ RESTART INTERPRETER
11.
12.  HERE COMPILE OK  2517 ,                  \ 5     0
13.                  45415 ,                  \ K     CR
14.                   3412 ,                  \ BELL  LF
15. 1 +BLOCK CONTINUED

BLOCK 501(765B)
 0. ( METAFORTH INTERPRETER WORDS                    HN/780319   2 OF ** )
 1. CODE  ABORT                  RENAME $BORT
 2.  HERE ' ABORT, !                    \ PROVIDE INDIRECT ABORT ADDR.
 3.  HERE ' ERROR, !                    \ PROVIDE INDIRECT ERROR ADDR.
 4.                 DP LDA,             \ ERROR MESSAGE ADDRESS
 5. DIRECT INITIALIZE                   \ INITIALIZE FORTH POINTERS
 6.                 SP0 LDB,  SP STB,           \ STACK POINTER SP
 7.                 RP0 LDB, -1 # ADB,          \ RESERVE LOC FOR IC
 8.                          RP STB,            \ RETURN STACK POINTER
 9.                 CLB,     SCR STB,          \ INPUT FROM CONSOLE
10.                        STATE STB,           \ EXECUTION MODE
11.                 ' ?+ 1+ # LDB,
12.                          RP ) STB,          \ SET IC TO ERROR-MSG
13.                 PUSH,                      \ MESSAGE ADDRESS
14. 1 +BLOCK CONTINUED
15.

BLOCK 502(766B)
 0. ( METAFORTH INTERPRETER WORDS                    JV/780330   3 OF ** )
 1. CODE ERROR   S) LDA, ERROR,
 2.
 3.  HERE
 4. : X ( EOT )   400 SWAP. !                \ CC=1 CH1=NULL
 5.               IMMEDIATE
 6.               TEG
 7.               LIT
 8.  HERE CONSTANT OK1
 9.               OK COUNT TYPE              \ TYPE OK CR BELL LF
10.               LIT 120 ,  EXPECT          \ REQUEST INPUT
11.               ;
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 503(767B)
 0. ( METAFORTH INTERPRETER WORDS                    JV/780119   4 OF ** )
 1. : ' ( TICK )  IMMEDIATE    RENAME $'
 2.               LIT 40 ,  WORD                    \ READ NEXT WORD
 3.               FIND                              \ GET ITS ADDRESS
 4.               LOC                               \ PUSH OR COMPILE IT
 5.               ;
 6.               LIT 17 , ERROR                    \ WHEN WORD NOT FOUND
 7. 2 +BLOCK ( 505D ) CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 505(771B)
 0. ( METAFORTH INTERPRETER                          JV/771007   1 OF ** )
 1. CR ." INTERPRETER " HERE .
 2. HERE GO !          \ SET START OF GO-SEQUENCE
 3. LIT  40 ,  WORD \ READ WORD FROM INPUT STRING
 4.
 5. FIND               \ LOCATE WORD IN DICTIONARY.
 6. EXECUTE            \ WHEN FOUND EXECUTE THE WORD
 7. 01                 \ AND RESTART THE INTERPRETER SEQUENCE
 8.
 9. NUMBER             \ OTHERWISE TRY TO CONVERT THE WORD TO A NUMBER
10. LITERAL            \ WHEN POSSIBLE PUSH OR COMPILE THE NUMBER
11. 01                 \ AND RESTART THE INTERPRETER SEQUENCE
12.
13. LIT 16 , ERROR  \ OTHERWISE ABORT THE INTERPRETING SEQUENCE
14.
15. 5 +BLOCK ( 510D ) CONTINUED

BLOCK 510(776B)
 0. ( METAFORTH MISCELLANEOUS                        HN/780319   1 OF ** )
 1. CR ." MISCELLANEOUS " HERE .
 2. : CODE                      RENAME $ODE
 3.               LIT 40 ,  WORD                    \ READ WORD
 4.               ENTER ASSEMBLER ;                 \ GENERATE A CODE ENTRY
 5.
 6. : (       IMMEDIATE         RENAME $(
 7.                   LIT 51 ,  WORD  ;             \ FOR COMMENTS
 8.
 9. CODE DEFINITIONS  CONTEXT LDA,
10.                   CURRENT STA,  NEXT,           \ SET DEFINITIONS VOC.
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 511(777B)
 0. ( METAFORTH MISCELLANEOUS                          HN/780313   2 OF ** )
 1.  CON SYSTEM     USER-AREA ,          \ ADDRESS OF SYSTEM AREA
 2.                 #CHF ,               \ CHARACTER OFFSET
 3.                 COMPUTER ,           \ COMPUTER SPECIFICATIONS
 4.                 COMPUTER 2 AND 0=    \ SOFTWARE MPY, DIV, ?
 5.                 IFTRUE
 6.                 MULTIPLY-DIVIDE ,    \ TRANSFER MPY AND DIV ADDRESS
 7.                 IFEND
 8.                 MASS-STORAGE ,       \ MASS STORAGE SPECIFICATIONS
 9.                 U1# ,                \ SAVE/RESTORE LENGTH OF USER
10.                 U2# ,                \ INDEXED      LENGTH OF USER
11. 11 +BLOCK ( 520D ) CONTINUED
12.
13.
14.
15.

BLOCK 520(1010B)
 0. ( METAFORTH SOFTWARE MPY DIV                       HN/780319   1 OF ** )
 1. COMPUTER 000002 AND IFTRUE  12 +BLOCK ( 530D ) CONTINUED
 2. IFEND CR ." SOFTWARE MPY, DIV, " HERE .
 3. 0 OCT T    0 ,  0 ,  0 ,  0 ,     \ FIVE WORDS TEMPORART STORAGE
 4. 1 +BLOCK CONTINUED
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 521(1011B)
 0. ( METAFORTH SOFTWARE MPY DIV                       HN/780319   2 OF ** )
 1. HERE 0 ,   MULTIPLY-DIVIDE !              \ MPY SUBROUTINE
 2. HERE 1- DUP
 3.       ) LDB, B ) LDB,                     \ GET OPERAND ADDRESS
 4.       ISZ,                                \ INCREMENT RETURN ADDR.
 5.       2120 ( CLE,SSA ) ,
 6.       3204 ( CMA,CME,INA ) ,
 7.       T STA,
 8.       B+,  7204 ( CMB,CME,INB ) ,
 9.       1500 ( ERA ) ,
10.       T 3 + STA,
11.       -20 # LDA,  T 4 + STA,
12.       CLA,  T 1+ STA,
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 522(10128)
 0. ( METAFORTH SOFTWARE MPY DIV                    HN/780319   3 OF ** )
 1.      HERE                                      \ START OF LOOP
 2.            66 ( CLE,ELA ) ,  T 2 + STA,
 3.            T 1+ LDA,  ELA,  T 1+ STA,
 4.            T 2 + LDA,
 5.            5250 ( RBL,CLE,SLB ) ,  T ADA,
 6.            E0,  T 1+ ISZ,
 7.            T 4 + ISZ,
 8.      JMP,
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 523(10138)
 0. ( METAFORTH SOFTWARE MPY DIV                    HN/780319   4 OF ** )
 1.      T 2 + STA,
 2.      T 3 + LDA,  T 1+ LDB,
 3.      A+,  CMB,
 4.      T 2 + LDA,
 5.      B=,  MULTIPLY-DIVIDE CP @ ) JMP,   \ RETURN
 6.      TCA,
 7.      A0, NOT   INB,
 8.      MULTIPLY-DIVIDE CP @ ) JMP,        \ RETURN
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 524(10148)
 0. ( METAFORTH SOFTWARE MPY DIV                    HN/780319   5 OF ** )
 1. HERE 0 ,  MULTIPLY-DIVIDE 1+ !                 \ DIV SUBROUTINE
 2. HERE 1- DUP
 3.      T STA,
 4.      ) LDA,  A ) LDA,                          \ GET OPERAND ADDRESS
 5.      ISZ,                                      \ INCREMENT RETURN ADDR.
 6.      T 4 + STA,
 7.      TCA,  T 2 + STA,
 8.      -21 # LDA,  T 1+ STA,
 9.      CLA,
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 525(1015B)
 0. ( METAFORTH SOFTWARE MPY DIV                    HN/780319   6 OF ** )
 1.     IF,                                         \ SKIP FIRST PART
 2.             HERE SWAP
 3.                 T LDA,
 4.                 66 ( CLE,ELA ) ,
 5.                 4026 ( ELB ) ,
 6.                 T STA,
 7.                 T 3 + LDA,  66 ( CLE,ELA ) ,
 8.     THEN,       T 2 + ADB,
 9.                 B-,  INA,
10.                 B+,  T 4 + ADB,
11.                 T 3 + STA,
12.                 T 1+ ISZ,
13.             JMP,
14.     MULTIPLY-DIVIDE 1+ CP ! ) JMP,              \ RETURN
15. 5 +BLOCK ( 530D ) CONTINUED

BLOCK 530(1022B)
 0. ( METAFORTH CONSOLE I/O                         HN/780319   1 OF ** )
 1. CR ." CONSOLE I/O " HERE .
 2. 1274 LOAD
 3.
 4.
 5.
 6.
 7. 12 +BLOCK ( 540D ) CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 540(1034B)
 0. ( METAFORTH MASS-STORAGE DEVICES                JV/780330   1 OF  1 )
 1. CR ." MASS-STORAGE I/O " HERE .
 2. MASS-STORAGE   377 AND   1 = IFTRUE 1130 ( 600D ) LOAD IFEND
 3. MASS-STORAGE   377 AND   3 = IFTRUE 1130 ( 600D ) LOAD IFEND
 4.
 5. 12 +BLOCK ( 550D ) CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 550(1046B)
 0. ( METAFORTH RESTART                                  JV/780316   1 OF ** )
 1. CR ." FORTH BASIC STARTS " HERE .
 2.
 3. HERE ' $ORTH 1+  #WDS 2 MAX + !          \ SET HEAD OF FORTH
 4.
 5. : GO  LIT 15 , $OAD ;
 6.
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 551(1047B)
 0. ( METAFORTH RESTART                                  JV/780320   2 OF ** )
 1. <CODE ZAP
 2. CR ." HOT START " HERE .
 3.          ' SYSTEM 1+ LDA,  U STA,        \ SET USER AREA ADDRESS
 4.          OK1 LDA,   INITIALIZE           \ START FORTH
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 552(1050B)
 0. ( METAFORTH RESTART                                  JV/780330   3 OF ** )
 1.  HERE                                  \ FOR COLD START
 2.          ERASE-CORE                    \ ZERO BUFFERS
 3.          ZAP                           \ RESTART FORTH
 4. COLD-START CP !  HERE COLD-START !     \ SET COLD-START ADDRESS
 5. CR ." COLD-START " HERE .
 6.          102004 # LDA,  4 STA,         \ LOC: 4    HLT 4
 7.          CLA,           5 STA,         \ LOC: 5    NOP !!!!!!!!
 8.          OWNER STA,                    \ RELEASE DISC
 9.          0 CLC,                        \ RESET ALL DEVICES
10.          0 STF,                        \ ACTIVATE INTERRUPT
11.          106706 # LDA,  6 # LDB,       \ GET ADDRESS
12.          HERE B ) STA,  INB,  INA,     \ SET CLC SC IN LOC. SC
13.               100 # BNE, RSS,          \ ALL INTERRUPT LOCS DONE?
14.          JMP,                          \ NO, DO NEXT ONE
15. 1 +BLOCK CONTINUED
```

```
BLOCK 553(1051B)
 0. ( METAFORTH RESTART                                  JV/780330   4 OF ** )
 1.          DUP IFTRUE    JSB,                  \ EXECUTE OTHER STARTS
 2.              OTHERWISE DROP
 3.              IFEND
 4. COMPUTER 000001 AND        IFTRUE            \ SET DMA INTERRUPT
 5.                  DMA 2 + # ) 14000 + # LDA, \ ( JSB DMA,I )
 6.                  6 STA,                      \ SET ON BASE PAGE
 7.          IFEND
 8. ( HERE ) # LDA,                              \ LOAD COLD START ADDRESS
 9.          RP ) STA,                           \ SET IC
10.          NEXT,
11. 7 +BLOCK ( 560D ) CONTINUED
12.
13.
14.
15.

BLOCK 560(1060B)
 0. ( METAFORTH POINTER INITIALIZATION           JV/780320   1 OF ** )
 1. CR ." POINTER INITIALIZATION " HERE .
 2. HERE MSG !                                   \ SET MSG BUFFER ADDRESS
 3. #2 @  #? !                                   \ SET CONSTANT AREA
 4. HERE 52 + DP  !                              \ RESERVE 84D CHARACTERS
 5. MEMORY RP0 !                                 \ SET BOTTOM OF RETURN
 6.                                              \ STACK
 7. MEMORY 1- RP !
 8. MEMORY RP-LENGTH - SET MEMORY                \ UPDATE MEMORY
 9. MEMORY SP0 !                                 \ SET BOTTOM OF STACK
10. MEMORY 1- SP !
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 561(1061B)
 0. ( METAFORTH POINTER INITIALIZATION           JV/780320   2 OF ** )
 1.   OTHER @ ! ASSEMBLER 1+ #WDS 2 MAX + ! \ HEAD OF ASSEMBLER
 2.   ' $ORTH 1+ ' ASSEMBLER
 3.                    2 + #WDS 2 MAX + ! \ LINK OF ASSEMBLER
 4.
 5.   ' $ORTH 1+ #WDS 2 MAX +   DUP          \ HEAD OF FORTH VOC.
 6.   CURRENT !  CONTEXT !                    \ SET VOC. POINTERS
 7.
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 562(1062B)
 0. ( METAFORTH POINTER INITIALIZATION          JV/780320   3 OF ** )
 1.
 2. CR ." GENERATED CODE " HERE .
 3. CR ." METAFORTH BINARY LENGTH " DP CP @ .
 4. CR CR CR
 5. ." END OF METAFORTH RUN" CR CR
 6.
 7. ;S
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 600(1130B)
 0. ( METAFORTH HP 7900A DISC I/O               HN/780212   1 OF 12 )
 1. CR ." HP 7900A DISC I/O " HERE .
 2.
 3. COMPUTER 000001 AND 0= IFTRUE CR ." NO DMA AVAILABLE "
 4.                                    ABORT IFEND
 5.
 6.          0 OCT DN                  \ DRIVE NUMBER 0-3
 7.          0 OCT ULP                 \ UPPER/LOWER PLATTER 0/1000
 8.          0 OCT CYLNO               \ CYLINDER #
 9.          0 OCT HDSEC               \ HEAD AND SECTOR NUMBER
10.      10000 # CONSTANT DWCMD        \ DISC WRITE COMMAND
11.      20000 # CONSTANT DRDCMD       \ DISC READ  COMMAND
12.      30000 # CONSTANT SKCMD        \ DISC SEEK  COMMAND
13. 120000 DC + # CONSTANT DCW1        \ DMA CONTROL WORD
14.
15. 1 +BLOCK CONTINUED

BLOCK 601(1131B)
 0. ( METAFORTH HP 7900A DISC I/O               HN/780313   2 OF 12 )
 1.
 2. MASS-STORAGE 000002 AND             \ CONTINUED PLATTERS ?
 3. IFTRUE   2 +BLOCK CONTINUED  IFEND
 4.
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 602(1132B)
 0. ( METAFORTH HP 7900A DISC I/O                    HN/780320   3 OF 12 )
 1.
 2. <CODE DADDR  \ CALCULATE DISC-ADDRESS
 3.              S) LDA,  -1 # ADA,  CLB,        \ BLK#-1
 4.              6 # DIV,                        \ A:=A/6, B:=REMAINDER
 5.              CAER,                           \ E:=BIT 0 OF A, A:=A/2
 6.              CYLNO STA,                      \ CYLINDER NUMBER
 7.              CLA,  ELA,  ALS,                \ HEAD # : 0/1
 8.              ULP IOR,                        \ ADD HEAD OFFSET
 9.              2RBL,                           \ REMAINDER * 4
10.              B IOR,  HDSEC STA,              \ HEAD/SECTOR ADDRESS
11.              POP,
12.
13. 3 +BLOCK CONTINUED
14.
15.

BLOCK 603(1133B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/771114   4 OF 12 )
 1.
 2. <CODE DADDR  \ CALCULATE DISC-ADDRESS
 3.              S) LDA,  -1 # ADA,  CLB,        \ BLK#-1
 4.              4604 # DIV,  S) STA,            \ SAVE BLK#/2436D
 5.              B LDA,  CLB,                    \ RESTORE REGISTERS
 6.              6 # DIV,                        \ A:=A/6, B:=REMAINDER
 7.              CAER,                           \ E:=BIT 0 OF A, A:=A/2
 8.              CYLNO STA,                      \ CYLINDER NUMBER
 9.              CLA,  ELA,  ALS,                \ HEAD # : 0/1
10.              2RBL,                           \ REMAINDER * 4
11.              B IOR,  HDSEC STA,              \ HEAD/SECTOR ADDRESS
12.
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 604(1134B)
 0. ( METAFORTH HP 7900A DISC I/O                    HN/780320   5 OF 12 )
 1.              ULP LDA,                        \ LOAD SYTEM HEADS
 2.              S) LDB,                         \ GET PLATTER OFFSET
 3.              B0,  CMA,  1000 # AND,          \ PLATTER ADDRESS
 4.              HDSEC ADM,                      \ HEAD/SECTOR ADDRESS
 5.              POP,
 6.
 7. 1000 ULP !                                   \ SELECT LOWERPLATTER
 8.                                              \ AS SYSTEM PLATTER
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 605(1135B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/780330   6 OF 12 )
 1. SUBROUTINE DCOM      \ OUTPUT COMMAND WORD TO SELECTED DRIVE
 2.                      DN ADB,  CC OTB,        \ OUTPUT COMMAND
 3.                      CC STC, CLF             \ ACTIVATE COMMAND
 4.                      HERE  DC SFS,  JMP,     \ FLAG ?
 5.                      RETURN
 6.
 7. SUBROUTINE STATUS    \ CHECK DISC STATUS
 8.                      HERE  CC SFS,  JMP,     \ WAIT FOR LAST ACTION
 9.                      6 STF,                  \ STOP DMA
10.                      DC CLC,  CC CLC,        \ RESET CONTROL SIGNALS
11.                      CLB,  DCOM              \ STATUS CHECK COMMAND
12.                      DC IIB,  4010 ( SIB ) ,  \ ANY ERROR?
13.                      44 HLT,                 \ ERROR, STATUS IN B
14.                      RETURN
15. 1 +BLOCK CONTINUED

BLOCK 606(1136B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/780330   7 OF 12 )
 1.
 2. SUBROUTINE SEEK      \ SEEK RECORD: DN CYLNO HDSEC
 3.                      DC CLC,  CC CLC,        \ RESET CONTROL SIG.
 4.                      CYLNO LDB,  DC OTB,     \ OUTPUT CYLNO
 5.                      DC STC, CLF             \ ACTIVATE COMMAND
 6.                      SKCMD LDB,  DCOM        \ ISSUE SEEK COMMAND
 7.                      HDSEC LDB,  DC OTB,     \ OUTPUT HDSEC
 8.                      DC STC, CLF             \ ACTIVATE COMMAND
 9.                      STATUS                  \ CHECK STATUS
10.                      RETURN
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 607(1137B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/771114   8 OF 12 )
 1.
 2. SUBROUTINE DSUB      \ DISC COMMAND IN B-REGISTER
 3.                      SEEK                    \ SEEK RECORD
 4.                      DC CLC,  CC CLC,        \ RESET CONTROL SIGNALS
 5.                      DCW1 LDB,  6 OTB,       \ DMA CONTROL WORD 1
 6.                      2 CLC,  S) IOR,         \ GENERATE NEXT C.W.
 7.                      2 OTA,  2 STC,          \ DMA CONTROL WORD 2
 8.                      BLOCK-LENGTH MINUS # LDB, \ LOAD NEXT C.W.
 9.                      2 OTB,                  \ DMA CONTROL WORD 3
10.                      RETURN
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 608(1140B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/780330   9 OF 12 )
 1.
 2. <CODE READ    \ READ BLOCK FROM DISC
 3.               0 ) # LDA,   DSUB              \ SET UP DMA FOR READ
 4.               DRDCMD LDA,  DN IOR,           \ GENERATE READ COMMAND
 5.               CC OTA,                        \ OUTPUT READ COMMAND
 6.               DC STC, CLF                    \ PREPARE DATA CHANNEL
 7.                6 STC, CLF                    \ ACTIVATE DMA
 8.               CC STC, CLF                    \ INITIATE READ
 9.               POP,
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 609(1141B)
 0. ( METAFORTH HP 7900A DISC I/O                    JV/780330  10 OF 12 )
 1.
 2. <CODE WRITE   \ WRITE BLOCK TO DISC
 3.               CLA,  DSUB                     \ SET UP DMA FOR WRITE
 4.               DC STC,  DC STF,               \ ENABLE DMA
 5.               6 STC, CLF                     \ ACTIVATE DMA
 6.               DWCMD LDA,  DN IOR,            \ GENERATE WRITE COMM.
 7.               CC OTA,                        \ OUTPUT WRITE COMMAND
 8.               CC STC, CLF                    \ INITIATE WRITE
 9.               POP,
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 610(1142B)
 0. ( METAFORTH HP 7900A DISC I/O                    HN/780320  11 OF 12 )
 1. <CODE ?DISC  STATUS  NEXT,                  \ CHECK DISC STATUS
 2.
 3. : DREAD      $MA ALLOCATE  DADDR  READ   ?DISC  $MA RELEASE ;
 4. : DWRITE     $MA ALLOCATE  DADDR  WRITE  ?DISC  $MA RELEASE ;
 5.
 6. : DISC       \ SET MASS-STORAGE TO DISC
 7.               RENAME $ISC
 8.               LIT $ROP    LIT ' BLOCK-MAP   1+ ,  $!
 9.               LIT DREAD   LIT ' BLOCK-READ  1+ ,  $!
10.               LIT DWRITE  LIT ' BLOCK-WRITE 1+ ,  $!
11.               ;
12.
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 611(1143R)
 0. ( METAFORTH HP 7900A DISC I/O                    HN/780212  12 OF 12 )
 1.
 2.     ' $ROP    ' BLOCK-MAP    1+ !
 3.     ' DREAD   ' BLOCK-READ   1+ !
 4.     ' DWRITE  ' BLOCK-WRITE  1+ !
 5.
 6. ;S
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 700(1274R)
 0. ( METAFORTH TTY CONSOLE I/O                      JV/780331   1 OF ** )
 1. COMPUTER 100000 AND                          \ INTERRUPT ?
 2. IFTRUE          2                            \ INTERRUPT DRIVER
 3. OTHERWISE       1                            \ NON-INTERRUPT DRIVER
 4. IFEND
 5.
 6.
 7. +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 701(1275R)
 0. ( METAFORTH TTY CONSOLE I/O : NON-INTER          JV/780331   2 OF ** )
 1. SUBROUTINE ITTY  TTY OTB,  TTY OTA,          \ OUTPUT 1 CHARACTER
 2.                  TTY STC, CLF                \ HIT CONSOLE
 3.                  HERE PAUSE,                 \ RETURN CONTROL
 4.                       TTY SFS,               \ WAIT UNTIL FLAG SET
 5.                  JMP,                        \ RETURN, FLAG NOT SET
 6.                  TTY LIA, CLF                \ GET CHARACTER
 7.                  177 # AND,                  \ ELIMINATE PARITY BIT
 8.                  RETURN
 9. 4 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 702(1276B)
 0. ( METAFORTH TTY CONSOLE I/O : INTERRUPT   JV/780331   3 OF ** )
 1.          0 ,                          \ TEMPORARY STORAGE A-REGISTER
 2.          0 ,                          \ TEMPORARY STORAGE TTY-USER
 3.          0 ,                          \ TEMPORARY STORAGE CHARACTER
 4. HERE     0 ,                          \ INTERRUPT ENTRY POINT
 5.          DUP 3 - STA,                 \ SAVE A-REGISTER
 6.          TTY LIA,                     \ GET CHARACTER
 7.          DUP  1- STA,                 \ SAVE CHARACTER
 8.          DUP 2 - ) LDA,               \ GET USER STATUS
 9.          177776 # AND,                \ CLEAR STATUS BITS, ACTIVATE
10.          DUP 2 - ) STA,               \ RESTORE USER STATUS
11.          DUP 3 - LDA,                 \ RESTORE A-REGISTER
12.          TTY CLC,                     \ CLEAR TTY CONTROL
13.          DUP ) JMP,                   \ RETURN FROM INTERRUPT
14. 1 +BLOCK CONTINUED
15.

BLOCK 703(1277B)
 0. ( METAFORTH TTY CONSOLE I/O : INTERRUPT   JV/780331   4 OF ** )
 1.          COLD-START CP @              \ COLD START ADDRESS
 2.          HERE COLD-START !            \ THIS IS NEW COLD START
 3. HERE     0 ,                          \ COLD START ENTRY POINT
 4.          SWAP DUP                     \ PREVIOUS COLD START ENTRY
 5.          IFTRUE    JSB,               \ DEFINED, SO EXECUTE
 6.          OTHERWISE DROP               \ NOT DEFINED
 7.          IFEND
 8.          OVER 114000 + # LDA,         \ GET 'JSB TTYINT,I'
 9.          TTY STA,                     \ STORE IN LOCATION TTY
10.          ) JMP,                       \ RETURN
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 704(1300B)
 0. ( METAFORTH TTY CONSOLE I/O : INTERRUPT   JV/780331   5 OF ** )
 1. SUBROUTINE 1TTY  TTY OTB,  TTY OTA,    \ OUTPUT 1 CHARACTER
 2.                  U LDA,  OVER 2 - STA, \ SAVE USER ADDRESS
 3.                  1 # LDA, U ) STA,     \ SET USER WAITING
 4.                  TTY STC, CLF          \ HIT CONSOLE
 5.                  PAUSE,                \ WAIT FOR INTERRUPT
 6.                  SWAP 1- LDA,          \ GET CHARACTER
 7.                  177 # AND,            \ ELIMINATE PARITY BIT
 8.                  RETURN
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.
```

```
BLOCK 705(13018)
 0. ( METAFORTH TTY CONSOLE I/O                    JV/780331   6 OF ** )
 1. 120000 # CONSTANT OTTY       \ TELETYPE OUTPUT CONTROL WORD
 2. 140000 # CONSTANT RTTY       \ TELETYPE INPUT  CONTROL WORD
 3. SUBROUTINE <TCH  OTTY LDB,  1TTY          \ OUTPUT CHARACTER
 4.              177 # ANE,                   \ BREAK CONDITION?
 5.              RETURN                       \ NO, CONTINUE PROGRAM
 6.              ABORT,                       \ YES, ABORT PROGRAM
 7. 1 +BLOCK CONTINUED
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 706(13028)
 0. ( METAFORTH TTY CONSOLE I/O                    JV/780331   7 OF ** )
 1. <CODE QUERY   HERE 1- ASK USER-AREA + ! \ UPDATE QUERY ADDRESS
 2.               S1) LDA,                  \ START OF MESSAGE BUF.
 3.               S) LDB,  A ADB,  S) STB,  \ END OF MESSAGE BUFFER
 4.               HERE OP STA,              \ SET OUTPUT BYTE POINTR
 5.                    HERE DUP
 6.                         RTTY LDA,       \ TTY CONTROL WORD
 7.                         1TTY            \ READ A CHARACTER
 8.                         A0, NOT  ABORT, \ NULL -> ABORT
 9.                         CH STA,         \ SAVE CHARACTER
10.                         10 # ANE,       \ BKSP?
11.                         IF, SWAP        \ NO
12.                              15 # ANE,  \ END OF TEXT? ( CR )
13.                              IF, SWAP   \ NO
14.                                  <TCH   \ ECHO CHARACTER
15. 1 +BLOCK CONTINUED

BLOCK 707(13038)
 0. ( METAFORTH TTY CONSOLE I/O                    JV/780403   8 OF ** )
 1.                              CH LDA,   \ GET CHARACTER
 2.                              DEPOSIT   \ IN MSG BUFFER
 3.                              S) LDA, TCA, \ = ( LAST WORD )
 4.                              OP ADA,      \ ADD BYTE POINTER
 5.                              A+,          \ BUFFER FULL?
 6.                    JMP,                   \ NO, GET NEXT CHAR.
 7.                         THEN,             \ EOT OR BUFFER FULL
 8.                         40 # LDA, <TCH \ ECHO BLANK
 9.                         40 # LDA,
10.                         DEPOSIT           \ BLANK INTO BUFFER
11.                         CLA, DEPOSIT      \ NULL INTO BUFFER
12.                         40 # LDA,
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 708(13048)
 0. ( METAFORTH TTY CONSOLE I/O                    JV/780331   9 OF ** )
 1.                          DEPOSIT          \ BLANK INTO BUFFER
 2.                          S1) LDA, IN STA, \ RESET IN
 3.                          2POP,            \ GO INTERPRET LINE
 4.                     THEN,                 \ BACKSPACE CHARACTER
 5.                     10 # LDA,  <TCH       \ ECHO BACKSPACE
 6.                     S1) LDA,  CMA,
 7.                     OP ADA,
 8.                     A+,                   \ POSITIVE # CHAR?
 9.                JMP,                       \ NO, IGNORE BKSP
10.                S1) ADA,                   \ YES, GET NEW IP
11.           JMP,                            \ SET NEW OP
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 709(13058)
 0. ( METAFORTH TTY CONSOLE I/O                    JV/780331  10 OF ** )
 1. <CODE TTY-OUT   HERE 1- TALK USER-AREA + !
 2.                 S) LDB,  TCB,  S) STB,  \ NEGATE CHAR. COUNT
 3.                 S1) LDA,  IP STA,       \ SET INPUT BYTE POINTER
 4.                 HERE FETCH              \ FETCH CHARACTER
 5.                      <TCH               \ TYPE CHARACTER
 6.                      S) ISZ,            \ ALL CHAR DONE?
 7.                 JMP,                    \ NO
 8.                 2POP,
 9. ;S
10.
11.
12.
13.
14.
15.

BLOCK 1000(17508)
 0. ( SECURITY                                     HN/780320   1 OF ** )
 1. FORTH DEFINITIONS
 2. BASE @  DECIMAL
 3. : SECURITY ;
 4. : ?COMP      @STATE 0= IF 16 ERROR THEN ;
 5. : ?EXEC      @STATE    IF 17 ERROR THEN ;
 6. : CPAIRS     -         IF 18 ERROR THEN ;
 7. : SCSP       SP@ CSP ! ;
 8. : CCSP       SP@ CSP @ - IF 19 ERROR THEN ;
 9. : ?LOADING   SCR @  0=   IF 23 ERROR THEN ;
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 1001(1751B)
 0. ( SECURITY                                          HN/780313   2 OF ** )
 1. :ORPHAN  ( ALREADY DEFINED ? )
 2.          WARNING @ ( NEEDED ? )
 3.          DUP 0< IF 22 ERROR THEN
 4.          IF 0 FIND DROP 1
 5.             IF CR HERE COUNT WRITE ." ALREADY DEFINED " THEN
 6.             CURRENT @ CONTEXT @ -
 7.             IF CR HERE COUNT WRITE ." CURRENT#CONTEXT " THEN
 8.             HERE @ [ OCTAL ] 400 [ DECIMAL ] *
 9.             IF 20 ERROR THEN
10.             SP@ HERE - 100 <
11.             IF 21 ERROR THEN
12.          THEN
13.          [ ' ENTER @ , ] ;
14.          ' ENTER !
15. 1 +BLOCK CONTINUED

BLOCK 1002(17525)
 0. ( SECURITY                                          JV/780330   3 OF ** )
 1. : DO      ?COMP           'C, DO     1 ; 1IMP DO
 2. : LOOP    ?COMP 1 CPAIRS  'C, LOOP     ; 1IMP LOOP
 3. : +LOOP   ?COMP 1 CPAIRS  'C, +LOOP    ; 1IMP +LOOP
 4. : IF      ?COMP           'C, IF     2 ; 1IMP IF
 5. : THEN    ?COMP 2 CPAIRS  'C, THEN     ; 1IMP THEN
 6. : ELSE    ?COMP 2 CPAIRS  'C, ELSE   2 ; 1IMP ELSE
 7. : BEGIN   ?COMP           'C, BEGIN  3 ; 1IMP BEGIN
 8. : END     ?COMP 3 CPAIRS  'C, END      ; 1IMP END
 9. : WHILE   ?COMP           'C, WHILE  4 ; 1IMP WHILE
10. : REPEAT  ?COMP 4 CPAIRS SWAP
11.                 3 CPAIRS 'C, REPEAT   ; 1IMP REPEAT
12. \ NOTE THAT THE PAIRING CONSTANTS ARE KEPT ON THE USERS STACK
13. \ ADOPT INSIDE LOOPS ETC. MUST BE CHANGED
14. 1 +BLOCK CONTINUED
15.

BLOCK 1003(1753B)
 0. ( SECURITY                                          JV/780320   4 OF ** )
 1. : :ORPHAN    ?EXEC          'C, :ORPHAN   SCSP ; 1IMP :ORPHAN
 2. : ORPHAN     ?EXEC          'C, ORPHAN    SCSP ; 1IMP ORPHAN
 3. : :          ?EXEC SCSP     'C, :          ; 1IMP :
 4. : ;          ?COMP CCSP     'C, ;          ; 1IMP ;
 5. : ;CODE      ?COMP CCSP     'C, ;CODE      ; 1IMP ;CODE
 6. : CODE       ?EXEC SCSP     'C, CODE       ; 1IMP CODE
 7. : ,CODE      ?EXEC          'C, ,CODE     SCSP ; 1IMP ,CODE
 8. : ADOPT      ?COMP 1 CSP +! 'C, ADOPT      ; 1IMP ADOPT
 9. : I          ?COMP LIT I    ,  ; 1IMP I
10. : J          ?COMP LIT J    ,  ; 1IMP J
11. : K          ?COMP LIT K    ,  ; 1IMP K
12. : LEAVE      ?COMP LIT LEAVE ,  ; 1IMP LEAVE
13. 1 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 1004(1754B)
 0. ( SECURITY                                          JV/780403   5 OF ** )
 1. : R>          ?COMP LIT R>    ,  ; 1IMP R>
 2. : >R          ?COMP LIT >R    ,  ; 1IMP >R
 3. : LIT         ?COMP LIT LIT   ,  ; 1IMP LIT
 4. : 'C,         ?COMP 'C, 'C,      ; 1IMP 'C,
 5. : VOCABULARY ?EXEC 'C, VOCABULARY ; 1IMP VOCABULARY
 6. : ;:          ?COMP CCSP LIT ;: ,  ; 1IMP ;:
 7. : ;S          ?EXEC ?LOADING R> DROP 'C, ;S   ; 1IMP ;S
 8. : CONTINUED   ?EXEC ?LOADING R> DROP 'C, CONTINUED ;
 9.                                      1IMP CONTINUED
10. : >B                  ?LOADING R> DROP 'C, >B   ; 1IMP >B
11. : +BLOCK              ?LOADING 'C, +BLOCK       ; 1IMP +BLOCK
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1005(1755B)
 0. ( SECURITY                                          JV/780320   5 OF ** )
 1. ASSEMBLER DEFINITIONS
 2. : EDOC        ?EXEC CCSP     'C, EDOC        ; 1IMP EDOC
 3. FORTH DEFINITIONS
 4. 1 WARNING !
 5. BASE ! ;S
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1020(1774B)
 0. ( MONITOR MULTIPROGRAMMER                           JV/780320   1 OF ** )
 1. FORTH DEFINITIONS
 2. : MONITOR ;
 3. BASE @ OCTAL
 4. ASSEMBLER
 5. HERE  0 ,                        \ MONITOR LOOP ENTERED HERE
 6.       DUP LDA,  U LDB,
 7.       2 # ADB,  B ) STA,    \ SAVE PROGRAM-COUNTER
 8.       SP  LDA,  2 # ADB,
 9.       B ) STA,                   \ SAVE STACK POINTER
10.       SYSTEM  5 +
11.       U1#     5 -
12.       1 +BLOCK LOAD              \ SAVE ALL PARAMETERS
13. 2 +BLOCK CONTINUED
14.
15.
```

```
BLOCK 1021(1775B)
 0. ( MONITOR MULTIPROGRAMMER                          HN/780315   2 OF ** )
 1.       OVER LDA,                  \ GET PARAMETER VALUE
 2.       INB,  B ) STA,             \ SAVE IT
 3.       1- SWAP 1+ SWAP
 4.       DUP
 5.       IFTRUE  0 +BLOCK CONTINUED  IFEND
 6.       2DROP
 7. ;S
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1022(1776B)
 0. ( MONITOR MULTIPROGRAMMER                          HN/780315   3 OF ** )
 1.       U   LDB,                   \ USER ADDRESS
 2.       HERE  INB,  B ) LDB,       \ NEXT USER
 3.             B ) LDA,             \ STATUS
 4.             A0,                  \ ACTIVE ?
 5.       JMP,                       \ NO, NEXT ONE
 6.       U   BNE,                   \ WAS LAST USER ?
 7.       DUP ) JMP,                 \ YES, ACTIVATE IMMEDIATELY
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1023(1777B)
 0. ( MONITOR MULTIPROGRAMMER                          HN/780315   4 OF ** )
 1.       U   STB,                   \ NO, ANOTHER USER
 2.       4 # ADB,
 3.       B ) LDA,  SP STA,          \ RESTORE STACK POINTER
 4.       INA,      SP1 STA,         \ RESTORE SP1
 5.       SYSTEM    5 +
 6.       U1#       5 -
 7.       1 +BLOCK LOAD              \ RESTORE ALL PARAMETERS
 8. 2 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 1024(200ØB)
 Ø. ( MONITOR MULTIPROGRAMMER                    HN/78Ø315   5 OF ** )
 1.        INB,  B ) LDA,          \ GET PARAMETER VALUE
 2.        OVER STA,               \ RESTORE IT
 3.        1- SWAP 1+ SWAP
 4.        DUP
 5.        IFTRUE  Ø +BLOCK CONTINUED  IFEND
 6.        2DROP
 7. ;S
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1025(2ØØ1B)
 Ø. ( MONITOR MULTIPROGRAMMER                    HN/78Ø32Ø   6 OF ** )
 1.        U   LDB,
 2.        2 # ADB,
 3.        B ) LDA,                \ GET PROGRAM COUNTER
 4.        A ) JMP,                \ RETURN TO CALLING PROGRAM
 5.
 6. ' PAUSE, @  1777 AND
 7.            !                   \ ACTIVATE MULTIPROGRAMMER
 8.
 9. FORTH
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.

BLOCK 1026(2ØØ2B)
 Ø. ( MONITOR MULTIPROGRAMMER OPERATOR          JV/78Ø32Ø   7 OF ** )
 1. Ø VARIABLE OPERATOR
 2. U1# U2# + 1- DP+!              \ ALLOCATE AREA
 3. SYSTEM OPERATOR U1# U2# + MOVE  \ COPY AREA
 4. OPERATOR DUP 1+ !              \ SET FOLLOWER
 5.
 6. : TEMPORARY TEG
 7.                OPERATOR
 8.                ASSEMBLER U FORTH
 9.                !
10.                ;
11.
12.
13. TEMPORARY
14. FORGET TEMPORARY
15. 1 +BLOCK  CONTINUED
```

```
BLOCK 1027(2003B)
 0. ( MONITOR MULTIPROGRAMMER                    JV/780330   8 OF ** )
 1.
 2. ORPHAN       HERE  -1 # LDA,
 3.                    U ) STA,
 4.                   PAUSE,
 5.              JMP,
 6.              EDOC
 7.
 8. : STOP  TEG  ADOPT  ;
 9.
10. : DEACTIVATE  GET  -1  SWAP  !  ;
11.
12. : REACTIVATE        0  SWAP  !  ;
13.
14. 1 +BLOCK CONTINUED
15.

BLOCK 1028(20048)
 0. ( MONITOR MULTIPROGRAMMER                    JV/780321   9 OF ** )
 1. : TASK  HERE                  \ DP
 2.         SWAP DP+! HERE        \ SP0
 3.         ROT  DP+! HERE        \ RP0
 4.         100000 VARIABLE       \ ENTRY & STATUS
 5.         HERE 1-               \ STATUS ADDRESS
 6.         ASSEMBLER !! FORTH @
 7.         1+ DUP @ ,            \ FOLLOWER
 8.         0 , 0 ,               \ PC & TIME-OUT
 9.         OVER SWAP !           \ UPDATE CHAIN
10.         0 , 0 ,               \  SP & RP
11.         ROT , SWAP ,          \ SP0 & RP0
12.         SWAP ,                \ DP
13. ( 1 +BLOCK CONTINUED
14.
15.

BLOCK 1029(2005B)
 0. ( MONITOR MULTIPROGRAMMER                    JV/780330  10 OF ** )
 1. 1        HERE U1# U2# + 11 +
 2.              DUP DP+!         \ ALLOCATE USER-AREA
 3.          SWAP 0 OVER !
 4.          DUP 1+ ROT 1- MOVE \ ZERO THE USER AREA
 5.          12 OVER 23 + !       \ BASE DECIMAL
 6.          ?F @ OVER ' ?F         @ + ! \ ERRORS
 7.          -1 OVER ' WARNING      @ + ! \ NO DEFINITIONS
 8.          1 OVER    ' BLK-LOW    @ + ! \ BLK-LOW
 9.          4604 OVER ' BLK-HIGH   @ + ! \ BLK-HIGH
10.          LIT 'C, STOP
11.          OVER OVER SWAP ' TYPE-IC    @ + !
12.          OVER OVER SWAP ' ACCEPT-IC  @ + !
13.          OVER OVER SWAP ' WRITE-IC   @ + !
14.          2DROP ;
15. 1 +BLOCK CONTINUED
```

```
BLOCK 1030(2006B)
 0. ( MONITOR MULTIPROGRAMMER                    JV/780330  11 OF ** )
 1. HERE ) STOP [
 2. : ACTIVATE 0 OVER !                          \ ACTIVATE
 3.             LIT [ ASSEMBLER ' NEXT, @ 1777 AND , ]
 4.             OVER 2 + !                       \ SET PC
 5.             DUP 6 + @  OVER 4 + !            \ EMPTY STACK
 6.             DUP 7 + @  1-
 7.             LIT ADOPT OVER !                 \ AUTO STOP AT END
 8.             1- R> OVER !                     \ SET IC
 9.             SWAP 5 + ! ;                     \ SET RP
10. FORTH
11. BASE ! ;S
12.
13.
14.
15.

BLOCK 1050(2032B)
 0. ( TIME BASE GENERATOR ROUTINES               HN/780202   1 OF  5 )
 1. FORTH DEFINITIONS
 2. ?DEF TBG 0= IFTRUE ." TBG DEVICE NOT DEFINED " ABORT IFEND
 3. BASE @  OCTAL
 4. 0 WARNING !   : CLOCK ;   1 WARNING !
 5. 0 VARIABLE TIME   0 ,                        \ TIME
 6. 0 ,                                          \ TICKS
 7. 1 , 1 ,                                      \ DELTA T ( MS)
 8.                                              \ & CONTROLWORD
 9. 0 , 0 , 0 ,                                  \ A,B,E&O REGISTERS
10. ASSEMBLER
11. HERE 0 ,                    \ START OF INTERRUPT ROUTINE
12.      TIME 2 + ISZ,  0 ,     \ INCREMENT TICKS
13.      TIME 1+ ISZ,           \ INCREMENT TIME
14.      RSS,                   \ SKIP WHEN NO CARRY
15. 1 +BLOCK CONTINUED

BLOCK 1051(2033B)
 0. ( TIME BASE GENERATOR ROUTINES               HN/780202   2 OF  5 )
 1.      TIME ISZ, 0 ,          \ ADD CARRY
 2.      TIME 5 + STA,          \ SAVE A-REG
 3.      TIME 6 + STB,          \ SAVE B-REG
 4.      1520 ( ERA,ALS ) ,     \ SAVE E-REG
 5.      1 SFC, INA,            \ SAVE O-REG
 6.      TIME 7 + STA,
 7.      U LDB,                 \ GET ADDRESS OF USER AREA .
 8.      HERE  3 # ADB,         \ TIME-OUT VALUE ADDRESS
 9.            B ) LDA,         \ TIME-OUT VALUE
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 1052(2034B)
 0. ( TIME BASE GENERATOR ROUTINES                HN/780320   3 OF  5 )
 1.              A0, NOT             \ SET ?
 2.              IF, B ) ISZ,        \ YES , INCREMENT VALUE
 3.                  IF,             \ VALUE ZERO, TIME-OUT !!!
 4.                   -3 # ADR,      \ STATUS ADDRESS
 5.                   B ) LDA,       \ GET STATUS
 6.                   0 ) # AND,     \ CLEAR BITS
 7.                   B ) STA,       \ RESTORE STATUS
 8.                   3 # ADR,       \ TIME-OUT ADDRESS
 9.                  THEN,
10.              THEN,
11.              -2 # ADR,           \ FOLLOWER
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1053(2035B)
 0. ( TIME BASE GENERATOR ROUTINES                HN/780202   4 OF  5 )
 1.              B ) LDB,            \ FOLLOWER STATUS ADDRESS
 2.              U BNE,              \ IS CURRENT USER ?
 3.              RSS,                \ YES, END OF CHAIN
 4.          JMP,                    \ CHECK NEXT USER
 5.          TIME 7 + LDA,
 6.          1 CLF, 36 ( SLA,ELA ) , \ RESTORE E
 7.          1 STF,                  \ RESTORE O
 8.          TIME 5 + LDA,           \ RESTORE A
 9.          TIME 6 + LDB,           \ RESTORE B
10.          TBG CLF,                \ RESET TIME BASE GENERATOR
11. DUP ) JMP,                       \ RETURN
12. # ) 14000 +  TBG !               \ SET JSB ,I IN LOC TBG
13. FORTH
14. 1 +BLOCK CONTINUED
15.

BLOCK 1054(2036B)
 0. ( TIME BASE GENERATOR ROUTINES                JV/780330   5 OF  5 )
 1. :ORPHAN  ABS TIME 3 + @ / 1+          \ TICKS TO WAIT
 2.          MINUS
 3.          77777 ASSEMBLER U FORTH @ ! \ DEACTIVATE USER
 4.          TIME-OUT !                  \ SET TIME-OUT
 5.          PAUSE ;
 6. ' MS  !                              \ CHANGE MS ROUTINE
 7. CODE START-CLOCK TIME 4 + LDA,       \ CONTROLWORD
 8.                  TBG OTA,            \ OUTPUT TO TBG .
 9.                  TBG STC, CLF        \ ACTIVATE TBG
10.                  NEXT,   EDOC
11. CODE STOP-CLOCK  TBG CLC,            \ STOP-CLOCK
12.                  NEXT,   EDOC
13. START-CLOCK
14. BASE ! ;S
15.
```

```
BLOCK 1100(2114B)
 0. ( PSEUDOVECTORS FOR 3 CHAR FORTH          HN/780320   1 OF  5 )
 1. FORTH DEFINITIONS
 2. #CHF 3 - IFTRUE ." PSEUDOVECTORS NOT DEFINED "
 3.                 ." FOR NON-STANDARD SYSTEMS "
 4.                 ABORT
 5.          IFEND
 6. BASE @  OCTAL  SECURITY
 7. : PV-DEFINE  PV-ADDRESS @ 0=
 8.              IF  HERE PV-ADDRESS !  20 DP+!  THEN  ;
 9. PV-DEFINE
10. : P-VX       HERE 2 + @ 7400 AND 400 /   \ INDEX
11.              PV-ADDRESS @ 0=
12.              IF  [ DECIMAL ]  25 ERROR  [ OCTAL ]
13.              THEN
14.              PV-ADDRESS @ + ;
15. 1 +BLOCK CONTINUED

BLOCK 1101(2115B)
 0. ( PSEUDOVECTORS FOR 3 CHAR FORTH          HN/780313   2 OF  5 )
 1. : ORCX  ?EXEC 'C, ASSEMBLER
 2.         HERE P-VX !
 3.         HERE 1+ , ;
 4.         1IMP ORCX
 5. : :ORX  ?EXEC SCSP ASSEMBLER EDOC FORTH
 6.         HERE P-VX !
 7.         LIT [ ' ORCX 1- @ , ] ,
 8.         HEAD @ LAST !
 9.         ] ;
10.         1IMP :ORX
11. : ADOX  ?COMP P-VX @ , ;   1IMP ADOX
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1102(2116B)
 0. ( PSEUDOVECTORS FOR 3 CHAR FORTH          HN/780313   3 OF  5 )
 1. ASSEMBLER DEFINITIONS
 2. : >>LX   HERE P-VX ! 0< ;
 3. : <<LX   P-VX @ HERE
 4.          OVER OVER XOR 76000 AND
 5.          IF  ASSEMBLER # ) FORTH
 6.          ELSE 101777 AND 2000 +
 7.          THEN
 8.          SWAP +!
 9.          ;
10. FORTH DEFINITIONS
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.
```

```
BLOCK 1103(2117B)
 0. ( PSEUDOVECTORS FOR 3 CHAR FORTH          JV/780313   4 OF  5 )
 1. ASSEMBLER
 2. ( STR )  HERE P-V1 !  2 # ADB,
 3.                       S) LDA,
 4.                       A ) STB,
 5.                       POP,
 6. ( RCL )  HERE P-V2 !  INB,
 7.                       B ) LDA,
 8.                       PUSH,
 9. : INTX   HERE P-VX !
10.          LIT ADO1 ,
11.          LIT ADO2 ,
12.          , ;   1IMP INTX
13. : STRX   ?COMP 'C, ADOX ;       1IMP STRX
14. : RCLX   ?COMP P-VX @ 1+ , ;    1IMP RCLX
15. 1 +BLOCK CONTINUED

BLOCK 1104(2120B)
 0. ( PSEUDOVECTORS FOR 3 CHAR FORTH          JV/780330   5 OF  5 )
 1. :OR0     HERE P-VX @ ! ;
 2. : <<CX   ADO0 HERE 1+ , ;
 3. : >>AX   ASSEMBLER >>LX , ;
 4. : <<:X   HEAD @ LAST ! ASSEMBLER EDOC FORTH
 5.          ADO0
 6.          LIT [ ' >>AX 1- @ , ] ,
 7.          ] ;
 8. ASSEMBLER DEFINITIONS
 9. : <>LX   HERE P-VX ! ;
10. : <L>X   P-VX @ ;
11. FORTH DEFINITIONS
12. BASE !  ;S
13.
14.
15.

BLOCK 1110(2126B)
 0. ( FILE-HANDLING                           HN/780313   1 OF  1 )
 1. FORTH DEFINITIONS
 2. BASE @ DECIMAL
 3. 0 WARNING !   : FILES ;   1 WARNING !
 4. 1 BLK-LOW !   2436 BLK-HIGH !        \ SYSTEM FILE
 5. : PHYSICAL  BLK-LOW @ + 1-           \ CONVERT TO PHYSICAL
 6.             DUP BLK-HIGH @ >         \ TOO HIGH ?
 7.             OVER BLK-LOW @ <         \ OR TOO LOW ?
 8.             OR IF 24 ERROR THEN ;
 9. 'C PHYSICAL DUP ' BLOCK ! ' BUFFER !
10. : FILE      HERE ROT , SWAP , ;: DUP @ BLK-LOW !
11.                                  1+ @ BLK-HIGH ! ;
12.  0001 2436 FILE SYSTEM-FILE
13. BASE ! ;S
14.
15.
```

```
BLOCK 1120(2140B)
 0, ( EDITOR                                          JV/780330   1 OF ** )
 1, FORTH DEFINITIONS
 2, VOCABULARY EDITOR  EDITOR DEFINITIONS  CHAIN FORTH
 3, BASE @ OCTAL
 4, : 'EOT' TEG ." OK" CR QUIT ;
 5, : T      CR LINEWRITE ;                \ <M> T     TYPES LINE M
 6, : E      LINE 20040 OVER !             \ <M> E     ERASES LINE M
 7,          DUP 1+ 37 MOVE
 8,          UPDATE ;
 9, : R      DUP E LINE BYTE               \ <M> R <TEXT><CR>
10,          0 WORD HERE COUNT 100 MIN \             REPLACES TEXT
11,          OVER + SWAP                   \         IN LINE M
12,          DO I B@ OVER B! 1+ LOOP
13,          DROP UPDATE
14,          'EOT' ;
15, 1 +BLOCK CONTINUED

BLOCK 1121(2141B)
 0, ( EDITOR                                          JV/780320   2 OF ** )
 1, : CL     LINE SWAP LINE SWAP           \ <N> <M> CL COPIES LINE N
 2,          40 MOVE                       \             TO M
 3,          UPDATE ;
 4, : D      DUP 17 <                      \ <M> D     DELETES LINE M
 5,          IF 17 SWAP
 6,             DO I 1+ I CL LOOP
 7,          ELSE DROP
 8,          THEN
 9,          17 E ;
10, : I      1+ DUP 17 <                   \ <M> I <TEXT><CR>
11,          IF DUP 16                     \             INSERT TEST
12,             DO I DUP 1+ CL -1 +LOOP \              AFTER LINE M
13,          THEN
14,          R ;
15, 1 +BLOCK CONTINUED

BLOCK 1122(2142B)
 0, ( EDITOR                                          HN/780320   3 OF ** )
 1, : L      BLK @ LIST ;
 2, : BMOVE DUP BLK !
 3,          SWAP BLOCK SWAP BUFFER
 4,          1000 MOVE ;
 5, :ORPHAN OVER BUFFER SWAP OVER !
 6,          DUP 1+ 777 MOVE
 7,          BLK ! ;
 8, DUP
 9, : ZERO  0 ADOPT ;
10, : CLEAR 20040 ADOPT ;
11, FORTH DEFINITIONS
12, : EDIT  'C, EDITOR LIST ;
13, EDITOR
14, BASE ! ;S
15,
```

```
BLOCK 1130(2152B)
 0. ( EDITOR                                          JV/780330   1 OF ** )
 1. FORTH DEFINITIONS
 2. VOCABULARY EDITOR  EDITOR DEFINITIONS  CHAIN FORTH
 3. BASE @ OCTAL
 4. : !2BL  20040 OVER !  DUP 1+ 37 MOVE ;
 5. : EMOVE 40 MOVE ;
 6. : TEXT  SP@ HERE -  140 <
 7.         IF [ DECIMAL ] 26 [ OCTAL ] ERROR THEN
 8.         HERE 50 + ;
 9. : STRING  HERE 1+ !2BL
10.          -1 IN +!
11.          WORD
12.          HERE 1+ TEXT EMOVE ;
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 1131(2153B)
 0. ( EDITOR                                          JV/780330   2 OF ** )
 1. : "      42 STRING ;
 2. : #      43 STRING ;
 3. : (      51 STRING ;
 4. : H   LINE TEXT EMOVE ;
 5. : T   CR DUP H LINEWRITE CR ;
 6. : R   LINE TEXT SWAP EMOVE UPDATE ;
 7. : D   DUP H  DUP 17 <
 8.       IF  17 SWAP
 9.           DO I 1+ LINE DUP 40 - EMOVE LOOP
10.       ELSE DROP
11.       THEN
12.       17 LINE !2BL
13.       UPDATE ;
14. 1 +BLOCK CONTINUED
15.

BLOCK 1132(2154B)
 0. ( EDITOR                                          JV/780330   3 OF ** )
 1. : L   BLK @ LIST ;
 2. : T   DUP 16
 3.       DO I LINE DUP 40 + EMOVE -1 +LOOP
 4.       1+ R ;
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 1133(2155B)
 0. ( EDITOR                                        JV/780330    4 OF ** )
 1. : BMOVE DUP BLK !
 2.          SWAP BLOCK SWAP BUFFER
 3.          1000 MOVE ;
 4. :ORPHAN OVER BUFFER SWAP OVER !
 5.          DUP 1+ 777 MOVE
 6.          BLK ! ;
 7. DUP
 8. ; ZERO  0 ADOPT ;
 9. : CLEAR 20040 ADOPT ;
10. FORTH DEFINITIONS
11. : EDIT  'C, EDITOR LIST ;
12. EDITOR
13. BASE ! ;S
14.
15.

BLOCK 1200(2260B)
 0. ( HP 7900 DISC I/O                              HN/780313    1 OF 11 )
 1. FORTH DEFINITIONS
 2. ?DEF DC 0= IFTRUE ." DISC SELECT CODE NOT DEFINED " ABORT IFEND
 3. COMPUTER 000001 AND 0= IFTRUE CR ." NO DMA AVAILABLE "
 4.                                                ABORT IFEND
 5. BASE @ OCTAL
 6.            0 VARIABLE DN                \ DRIVE NUMBER 0-3
 7.            0 VARIABLE ULP               \ UPPER/LOWERPLATTER 0/1000
 8.            0 VARIABLE CYLNO             \ CYLINDER NUMBER
 9.            0 VARIABLE HDSEC             \ HEAD AND SECTOR NUMBER
10.         1000 CONSTANT BLOCK-LENGTH      \ LENGTH OF BLOCK
11.        10000 VARIABLE DWCMD             \ DISC WRITE COMMAND
12.        20000 VARIABLE DRDCMD            \ DISC READ COMMAND
13.        30000 VARIABLE SKCMD             \ DISC SEEK COMMAND
14. 120000 DC + VARIABLE DCW1               \ DMA CONTROL WORD
15. 1 +BLOCK CONTINUED

BLOCK 1201(2261B)
 0. ( HP 7900 DISC I/O                              HN/780203    2 OF 11 )
 1.
 2. ?DEF CONTINUED-PLATTERS                 \ CONTINUED PLATTERS ?
 3. IFTRUE    2 +BLOCK CONTINUED  IFEND
 4.
 5. 1 +BLOCK CONTINUED
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
```

```
BLOCK 1202(2262B)
 0. ( HP 7900 DISC I/O                                  JV/780320   3 OF 11 )
 1.
 2. CODE DADDR  \ CALCULATE DISC-ADDRESS
 3.              S) LDA,  -1 # ADA,  CLB,      \ BLK#-1
 4.              6 # DIV,                      \ A:=A/6, B:=REMAINDER
 5.              CAER,                         \ E:=BIT 0 OF A, A:=A/2
 6.              CYLNO STA,                    \ CYLINDER NUMBER
 7.              CLA,  ELA,  ALR,              \ HEAD # : 0/1
 8.              ULP IOR,                      \ ADD HEAD OFFSET
 9.              2RBL,                         \ REMAINDER * 4
10.              B IOR,  HDSEC STA,            \ HEAD/SECTOR ADDRESS
11.              POP,   EDOC
12.
13. 3 +BLOCK CONTINUED
14.
15.

BLOCK 1203(2263B)
 0. ( HP 7900 DISC I/O                                  HN/780203   4 OF 11 )
 1.
 2. CODE DADDR  \ CALCULATE DISC-ADDRESS
 3.              S) LDA,  -1 # ADA,  CLB,      \ BLK#-1
 4.              4604 # DIV,  S) STA,          \ SAVE BLK#/2436D
 5.              B LDA,  CLB,                  \ RESTORE REGISTERS
 6.              6 # DIV,                      \ A:=A/6, B:=REMAINDER
 7.              CAER,                         \ E:=BIT 0 OF A, A:=A/2
 8.              CYLNO STA,                    \ CYLINDER NUMBER
 9.              CLA,  ELA,  ALR,              \ HEAD # : 0/1
10.              2RBL,                         \ REMAINDER * 4
11.              B IOR,  HDSEC STA,            \ HEAD/SECTOR ADDRESS
12.
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 1204(2264B)
 0. ( HP 7900 DISC I/O                                  HN/780203   5 OF 11 )
 1.              ULP LDA,                      \ LOAD SYTEM HEADS
 2.              S) LDB,                       \ GET PLATTER OFFSET
 3.              B0,  CMA,  1000 # AND,        \ PLATTER ADDRESS
 4.              HDSEC ADM,                    \ HEAD/SECTOR ADDRESS
 5.              POP,   EDOC
 6.
 7. 1000 ULP !                                 \ SELECT LOWERPLATTER
 8.                                            \ AS SYSTEM PLATTER
 9.
10. 1 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 1205(2265B)
 0. ( HP 7900 DISC I/O                                  JV/780330   6 OF 11 )
 1. SUBROUTINE DCOM     \ OUTPUT COMMAND WORD TO SELECTED DRIVE
 2.                     DN ADR,  CC OTB,      \ OUTPUT COMMAND
 3.                     CC STC, CLF           \ INITIATE COMMAND
 4.                     HERE  DC SFS,  JMP,   \ FLAG ?
 5.                     DCOM ) JMP,     EDOC
 6.
 7. SUBROUTINE STATUS   \ CHECK DISC STATUS
 8.                     HERE  CC SFS,  JMP,   \ WAIT FOR LAST ACTION
 9.                     6 STF,                \ STOP DMA
10.                     DC CLC,  CC CLC,      \ RESET CONTROL SIGNALS
11.                     CLB,  DCOM JSB,       \ STATUS V CHECK COMMAND
12.                     DC LIB,  4010 ( SLB ) ,  \ ANY ERROR?
13.                     44 HLT,               \ ERROR, STATUS IN B
14.                     STATUS ) JMP,     EDOC
15. 1 +BLOCK CONTINUED

BLOCK 1206(2266B)
 0. ( HP 7900 DISC I/O                                  JV/780330   7 OF 11 )
 1.
 2. SUBROUTINE SEEK     \ SEEK RECORD: DN CYLNO HDSEC
 3.                     DC CLC,  CC CLC,      \ RESET CONTROL SIG.
 4.                     CYLNO LDB,  DC OTB,   \ OUTPUT CYLNO
 5.                     DC STC, CLF           \ ACTIVATE
 6.                     SKCMD LDB,  DCOM JSB, \ SEEK COMMAND
 7.                     HDSEC LDB,  DC OTB,   \ OUTPUT HDSEC
 8.                     DC STC, CLF           \ ACTIVATE
 9.                     STATUS JSB,           \ CHECK STATUS
10.                     SEEK ) JMP,     EDOC
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1207(2267B)
 0. ( HP 7900 DISC I/O                                  HN/780203   8 OF 11 )
 1.
 2. SUBROUTINE DSUB     \ DISC COMMAND IN B-REGISTER
 3.                     SEEK JSB,             \ SEEK RECORD
 4.                     DC CLC,  CC CLC,      \ RESET CONTROL SIGNALS
 5.                     DCW1 LDB,  6 OTB,     \ DMA CONTROL WORD 1
 6.                     2 CLC,  S) IOR,       \ GENERATE NEXT C.W.
 7.                     2 OTA,  2 STC,        \ DMA CONTROL WORD 2
 8.                     BLOCK-LENGTH MINUS # LDB, \ LOAD NEXT C.W.
 9.                     2 OTB,                \ DMA CONTROL WORD 3
10.                     DSUB ) JMP,     EDOC
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 1208(2270B)
 0. ( HP 7900 DISC I/O                                  JV/780330   9 OF 11 )
 1.
 2. CODE DREAD    \ READ BLOCK FROM DISC
 3.               0 ) # LDA,  DSUB JSB,          \ SET UP DMA FOR READ
 4.               DRDCMD LDA,  DN IOR,           \ GENERATE READ COMMAND
 5.               CC OTA,                        \ OUTPUT READ COMMAND
 6.               DC STC, CLF                    \ PREPARE DATA CHANNEL
 7.                6 STC, CLF                    \ ACTIVATE DMA
 8.               CC STC, CLF                    \ INITIATE READ
 9.               POP,    EDOC
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 1209(2271B)
 0. ( HP 7900 DISC I/O                                  JV/780330  10 OF 11 )
 1.
 2. CODE DWRITE   \ WRITE BLOCK TO DISC
 3.               CLA,  DSUB JSB,                \ SET UP DMA FOR WRITE
 4.               DC STC,  DC STF,               \ ENABLE DMA
 5.               6 STC, CLF                     \ ACTIVATE DMA
 6.               DWCMD LDA,  DN IOR,            \ GENERATE WRITE COMM.
 7.               CC OTA,                        \ OUTPUT WRITE COMMAND
 8.               CC STC, CLF                    \ INITIATE WRITE
 9.               POP,    EDOC
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 1210(2272B)
 0. ( HP 7900 DISC I/O                                  HN/780320  11 OF 11 )
 1. CODE ?DISC  STATUS JSB, NEXT,  EDOC         \ CHECK DISC STATUS
 2.
 3. : DREAD      DMA ALLOCATE  DADDR  DREAD   ?DISC  DMA RELEASE ;
 4. : DWRITE     DMA ALLOCATE  DADDR  DWRITE  ?DISC  DMA RELEASE ;
 5.
 6. : DISC       \ SET MASS-STORAGE TO DISC
 7.               LIT 'C,  DROP    ' BLOCK-MAP   !
 8.               LIT 'C,  DREAD   ' BLOCK-READ  !
 9.               LIT 'C,  DWRITE  ' BLOCK-WRITE !
10.               ;
11.
12. BASE ! ;S
13.
14.
15.
```

```
BLOCK 1290(2412B)
 0. ( STERREWACHT UTRECHT CONFIGURATION          JV/780403   1 OF  1 )
 1. BASE @ OCTAL
 2.          11 CONSTANT TBG                    \ TIME BASE GENERATOR
 3.          12 CONSTANT DC                     \ 7960 DISC DATA CH.
 4.          13 CONSTANT CC                     \ 7900 DISC COMMAND CH.
 5.          14 CONSTANT TTY                    \ B100 CONSOLE
 6.          17 CONSTANT VDU                    \ TELETYPE TERMINAL
 7. BASE ! ;S
 8.
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1300(2424B)
 0. ( LINE PRINTER DRIVER                       HN/780321   1 OF ** )
 1. FORTH DEFINITIONS
 2. ?DEG LP  0=  IFTRUE ." PRINTER NOT IN SYSTEM " ABORT IFEND
 3. BASE @ OCTAL
 4. 0 VARIABLE LP-USER                          \ PRINTER BUSY FLAG
 5. COMPUTER  100000 AND
 6. IFTRUE    2                                 \ INTERRUPT DRIVER
 7. OTHERWISE 1                                 \ NON-INTERRUPT DRIVER
 8. IFEND
 9.
10.
11.
12.   +BLOCK CONTINUED
13.
14.
15.

BLOCK 1301(2425B)
 0. ( LINE PRINTER DRIVER : NON-INTERRUPT       JV/780330   2 OF ** )
 1.
 2.
 3. CODE 1PRINT      S) LDA,  LP OTA,  LP STC, CLF
 4.                  HERE PAUSE,  LP SFS,  JMP,
 5.                  LP CLC,
 6.                  POP,
 7.                  ENDC
 8.
 9.
10. 2 +BLOCK CONTINUED
11.
12.
13.
14.
15.
```

```
BLOCK 1302(2426B)
 0. ( LINE PRINTER DRIVER : INTERRUPT          JV/780330   3 OF ** )
 1. ASSEMBLER   0 ,                  \ TEMPORARY STORAGE A-REGISTER
 2. HERE  #     114000 + LP !        \ SET INTERRUPT LOCATION
 3. HERE  0 ,   LP CLC,              \ DEACTIVATE PRINTER
 4.             DUP 1- STA,          \ SAVE A-REGISTER
 5.             LP-USER ) LDA,
 6.             177776 # AND,
 7.             LP-USER ) STA,       \ REACTIVATE USER
 8.             DUP 1- LDA,          \ RESTORE A-REGISTER
 9.             ) JMP,               \ RETURN FROM INTERRUPT ROUTINE
10. CODE 1PRINT S) LDA,  LP OTA,     \ OUTPUT CHARACTER
11.              1 # LDA, U ) STA,   \ DEACTIVATE USER
12.              LP STC, CLF         \ HIT LINE PRINTER
13.              PAUSE,              \ RELEASE CONTROL
14.              POP,   ENDC
15. 1 +BLOCK CONTINUED

BLOCK 1303(2427B)
 0. ( LINE PRINTER DRIVER                      JV/780403   4 OF ** )
 1. : PRINT    LP-USER ALLOCATE  0
 2.            DO DUP B@ 1PRINT 1+ LOOP
 3.            DROP
 4.            LP-USER RELEASE ;
 5.
 6. : PRINTER  LIT 'C, PRINT WRITE-IC ! ;
 7.
 8. BASE ! ;S
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1310(2436B)
 0. ( VDU DRIVER                               HN/780320   1 OF ** )
 1. FORTH DEFINITIONS
 2. ?DEF VDU 0=  IFTRUE ." VDU      NOT IN SYSTEM " ABORT IFEND
 3. BASE ! OCTAL
 4. 0 VARIABLE VDU-USER                        \ VDU BUSY FLAG
 5. COMPUTER  100000 AND
 6. IFTRUE    2                                \ INTERRUPT DRIVER
 7. OTHERWISE 1                                \ NON-INTERRUPT DRIVER
 8. IFEND
 9.
10.
11.
12.    +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 1311(2437B)
  0. ( VDU DRIVER   NON-INTERRUPT                    JV/780330   2 OF ** )
  1. CODE 1VDU      120000 # LDA,  VDU OTA,    \ OUTPUT WRITE CTRL WORD
  2.                S) LDA,  VDU OTA,          \ OUTPUT CHARACTER
  3.                VDU STC, CLF               \ HIT VDU
  4.                HERE PAUSE, VDU SFS, JMP,  \ WAIT FOR FLAG
  5.                POP,                       \ POP CHARACTER
  6.                EDOC
  7.
  8. 3 +BLOCK CONTINUED
  9.
 10.
 11.
 12.
 13.
 14.
 15.

BLOCK 1312(2440B)
  0. ( VDU DRIVER   INTERRUPT                      HN/780320   3 OF ** )
  1. ASSEMBLER      0 ,                        \ TEMP. STORAGE A-REG.
  2. HERE # 114000  +  VDU !                   \ SET INTERRUPT LOCATION
  3. HERE           0 ,
  4.                DUP 1- STA,                \ SAVE A-REGISTER
  5.                VDU CLC,                   \ DEACTIVATE VDU
  6.                VDU-USER ) LDA,            \ GET USER STATUS
  7.                177776 # AND,              \ CLEAR VDU-BIT
  8.                VDU-USER ) STA,            \ REACTIVATE USER
  9.                DUP 1- LDA,                \ RESTORE A-REGISTER
 10.                ) JMP,                     \ RETURN FROM INTERRUPT
 11.                                           \ SUBROUTINE
 12.
 13. 1 +BLOCK CONTINUED
 14.
 15.

BLOCK 1313(2441B)
  0. ( VDU DRIVER   INTERRUPT                      JV/780330   4 OF ** )
  1. CODE 1VDU      1 # LDA,  U ) STA,         \ DEACTIVATE USER
  2.                120000 # LDA,  VDU OTA,    \ OUTPUT WRITE CTRL WORD
  3.                S) LDA,  VDU OTA,          \ OUTPUT CHARACTER
  4.                VDU STC, CLF               \ HIT VDU
  5.                PAUSE,                     \ RELEASE CONTROL
  6.                POP,                       \ POP CHARACTER
  7.                EDOC
  8.                                           \ SUBROUTINE
  9.
 10. 1 +BLOCK CONTINUED
 11.
 12.
 13.
 14.
 15.
```

```
BLOCK 1314(2442B)
 0. ( VDU DRIVER                                        JV/780403   5 OF ** )
 1. : VDU-WRITE   VDU-USER ALLOCATE  0
 2.               DO DUP 80 1VDU 1+ LOOP
 3.               DROP
 4.               VDU-USER RELEASE ;
 5.
 6. : DISPLAY     LIT 'C, VDU-WRITE WRITE-IC ! ;
 7.
 8. BASE !     ;S
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1400(2570B)
 0. ( 32-BIT OPERATORS                                  HN/780315   1 OF  4 )
 1. FORTH DEFINITIONS
 2. : 32BIT ;
 3. : 2CONSTANT CONSTANT . ;CODE
 4.                        SP LDA, -1 # ADA, SP STA, \ RESERVE LOC.
 5.                        INB,  B ) LDA,    \ GET MOST SIGN. PART
 6.                        INB,  B ) LDB,    \ GET LEAST SIGN. PART
 7.                        S) STB,           \ PUSH LEAST SIG. PART
 8.                        PUSH,             \ PUSH MOST SIGN. PART
 9.                        EDOC
10.
11. : 2VARIABLE VARIABLE , ;
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1401(2571B)
 0. ( 32-BIT OPERATORS                                  HN/780315   2 OF  4 )
 1. CODE 2@      S) LDB,  B ) LDA,          \ GET M.S.P.
 2.              INB,     B ) LDB,          \ GET L.S.P.
 3.              S) STB,                    \ PUT L.S.P.
 4.              PUSH,                      \ PUSH M.S.P.
 5.              EDOC
 6.
 7. CODE 2!      S) LDB,                    \ GET ADDRESS
 8.              S1) LDA, B ) STA,          \ STORE M.S.P.
 9.              ISP,     INB,
10.              S1) LDA, B ) STA,          \ STORE L.S.P.
11.              2POP,
12.              EDOC
13.
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 1402(2572B)
 0. ( 32-BIT OPERATORS                                  HN/780315   3 OF  4 )
 1. CODE 2SWAP  SP LDA,  2 # ADA,  N STA,
 2.             S) LDA,  N ) LDB,  N ) STA,  S) STB, \ SWAP M.S.P.
 3.             N  ISZ,
 4.             S1) LDA, N ) LDB,  N ) STA, S1) STB, \ SWAP L.S.P.
 5.             NEXT,
 6.             EDOC
 7.
 8. CODE 2DUP   S) LDA,                                \ GET M.S.P.
 9.             SP LDB,  -1 # ADB,  SP STB,
10.             S1) LDB, S)  STB,                      \ COPY L.S.P.
11.             PUSH,
12.             EDOC
13.
14. 1 +BLOCK CONTINUED
15.

BLOCK 1403(2573B)
 0. ( 32-BIT OPERATORS                                  HN/780321   4 OF  4 )
 1. CODE 2OVER  SP LDB,  -1 # ADB,  SP STB,  \ RESERVE ONE LOC.
 2.             3 # ADB,  B ) LDA,           \ GET M.S.P.
 3.             INB,      B ) LDB,           \ GET L.S.P.
 4.             S) STB,                      \ PUSH L.S.P.
 5.             PUSH,                        \ PUSH M.S.P.
 6.             EDOC
 7.
 8. : 2ROT      ASSEMBLER N 2! 2SWAP N 2@ 2SWAP FORTH ;
 9.
10. ;S
11.
12.
13.
14.
15.

BLOCK 1500(2734B)
 0. ( MISCELLANEOUS                                     JV/780403   1 OF ** )
 1. FORTH DEFINITIONS
 2. : ARRAY   0 CONSTANT DP+! ;CODE
 3.                          INB,  S) ADB,  S) STB,  NEXT,  EDOC
 4. : 2ARRAY  0 CONSTANT BYTE 1+ DP+! ;CODE
 5.                          INB,  S) ADB,  S) ADB,  S) STB,
 6.                        NEXT,   EDOC
 7. : WHERE   HEAD @ ID.   DUP ." BLOCK " T.
 8.           ." BASE " BASE @ T. ;
 9. : MYSELF  LAST @
10.           [ #CHF IFTRUE ]     LIT [ #CHF 3 + 2/ , ]
11.                [ OTHERWISE ] DUP COUNT SWAP DROP
12.                              3 MAX 3 + 2/
13.                [ IFEND ] + , ;      ;IMP MYSELF
14. 1 +BLOCK CONTINUED
15.
```

```
BLOCK 1501(2735B)
 0. ( MISCELLANEOUS                                   HN/780315   2 OF ** )
 1. : CYCLF   0 CONSTANT , ;CODE
 2.                              INB,  B ) LDA,
 3.                              B ) ISZ,
 4.                              INB,
 5.                              B ), ANE,  RSS,  PUSH,
 6.                              -1 # ADB,
 7.                              CLA,  B ) STA,
 8.                              B ) ISZ,
 9.                              PUSH,
10.                              EDOC
11.
12. 1 *BLOCK CONTINUED
13.
14.
15.

BLOCK 1502(2736B)
 0. ( MISCELLANEOUS                                   JV/780320   3 OF ** )
 1. : CLIST   CR PAGE 1+ SWAP
 2.           DO I BLOCK @
 3.              IF I . SPACE
 4.                 I BLK !
 5.                 @ LINEWRITE
 6.                 CR
 7.              THEN
 8.           LOOP ;
 9.
10.
11. 1 *BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 1503(2737B)
 0. ( MISCELLANEOUS                                   HN/780315   4 OF ** )
 1. : BLIST   PAGE 1+ SWAP @ ROT ROT
 2.           DO I BLOCK @
 3.              IF I LIST 1+
 4.                 DUP 3 =
 5.                 IF PAGE DROP 0 THEN
 6.              THEN
 7.           LOOP
 8.           DROP ;
 9.
10. ;S
11.
12.
13.
14.
15.
```

```
BLOCK 1600(3100B)
 0. ( XY STORAGE DISPLAY                                  JV/771010   1 OF 12 )
 1.
 2. FORTH DEFINITIONS
 3. ?DEF XYS 0= IFTRUE CR ." NO STORAGE DISPLAY DEFINED " IFEND
 4. ?DEF V+  0= IFTRUE CR ." NO ARITHMETIC PACKAGE "      IFEND
 5.
 6. VOCABULARY STORAGE-DISPLAY
 7. STORAGE-DISPLAY DEFINITIONS
 8. CHAIN FORTH
 9.
10. BASE @ OCTAL
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1601(3101B)
 0. ( XY STORAGE DISPLAY                                  HN/780315   2 OF 12 )
 1.
 2. \ AUTO INDEXING FOR STORAGE DISPLAY
 3. 0     VARIABLE (S)
 4. 0 (S) SET       X
 5. 1 (S) SET       Y
 6.
 7. : VECTOR  0 CONSTANT  -1 DP+!  !CODE
 8.                                  INB,  (S) ADB,
 9.                                  B LDA,  PUSH,
10.                                  EDOC
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1602(3102B)
 0. ( XY STORAGE DISPLAY                                  JV/780330   3 OF 12 )
 1. 2 ,CODE PAGE  0 ( STORAGE-USER ) ,  0 ( A-REGISTER ) ,
 2.               U LDA,  ' PAGE STA,          \ SAVE USER ADDRESS
 3.               0 ) # LDB,  U ) STB,         \ DEACTIVATE USER
 4.               XYS STC, CLF                 \ CLEAR SCREEN
 5.               PAUSE,                       \ RETURN
 6.               NEXT,
 7. HERE # 114000 + XYS !                      \ SET INTERRUPT JSB
 8. HERE  0 ,  XYS CLC,                        \ CLEAR CONTROL FLAG
 9.            ' PAGE 1+ STA,                  \ SAVE A-REGISTER
10.            ' PAGE ) LDA,  77777 # AND,     \ MASK OFF BIT 15
11.            ' PAGE ) STA,                   \ REACTIVATE USER
12.            ' PAGE 1+ LDA,                  \ RESTORE A-REGISTER
13. ) JMP,                                     \ RETURN FROM INTERRUPT
14. EDOC
15. 1 +BLOCK CONTINUED
```

```
BLOCK 1603(3103B)
 0. ( XY STORAGE DISPLAY                                  JV/771010   4 OF 12 )
 1.
 2. CODE SHOW  \ SET DOT AT COORDINATE  (Y,X)
 3.             S1) LDA,  AL8,  S) TOR,          \ X,Y COORDINATE
 4.             XYS OTA,                         \ OUTPUT COORDINATES
 5.             2POP,
 6. FDOC
 7.
 8. 1 +BLOCK CONTINUED
 9.
10.
11.
12.
13.
14.
15.

BLOCK 1604(3104B)
 0. ( XY STORAGE DISPLAY                                  JV/771010   5 OF 12 )
 1.
 2. 0 VARIABLE ?POS                  \ ON/OFF SCREEN SWITCH
 3. 0 ,                              \ X-POSITION
 4. 0 ,                              \ Y-POSITION
 5.
 6. 0 ?POS SET OFF
 7. 1 ?POS SET ON
 8.
 9. 1 +BLOCK CONTINUED
10.
11.
12.
13.
14.
15.

BLOCK 1605(3105B)
 0. ( XY STORAGE DISPLAY                                  JV/771010   6 OF 12 )
 1. : DRAW  \ A VECTOR FROM CURRENT POSITION TO (Y,X)
 2.          ?POS @                                  \ DARK VECTOR?
 3.          IF   2DUP  ?POS 1+ D@  V-               \ DIRECTION VECTOR
 4.               OVER ABS  OVER ABS  MAX            \ # POINTS TO SHOW
 5.               DUP  1+ 0
 6.               DO   I 2OVER ROT
 7.                    4 PICK  V*/
 8.                    ?POS 1+ D@  V+                \ POINT TO SHOW
 9.                    SHOW                          \ POINT ON SCREEN
10.               LOOP 3 NDROP
11.          THEN
12.          ?POS 1+ D!                              \ SET ACTUAL POSITION
13.          ;
14. : +DRAW ?POS 1+ D@ V+ DRAW ;                     \ RELATIVE VECTOR
15. 1 +BLOCK CONTINUED
```

```
BLOCK 1606(3106B)
 0. ( XY STORAGE DISPLAY                              JV/771010   7 OF 12 )
 1.
 2. DECIMAL
 3. VECTOR ORG        0 ,      0 ,
 4. VECTOR RNG     1000 ,   1000 ,
 5. VECTOR CORNER     0 ,      0 ,
 6. VECTOR FRAME    255 ,    255 ,
 7. VECTOR INC     -100 ,   -100 ,
 8. VECTOR DASH      10 ,     10 ,
 9.
10. 3 POLY /10       20 ,  25 ,  20 ,
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.

BLOCK 1607(3107B)
 0. ( XY STORAGE DISPLAY                              JV/771010   8 OF 12 )
 1.
 2. : SIZE  \ SET SCALE FACTORS FOR THE AXES
 3.         OVER - DUP  RNG !
 4.         0 ' /10 !                                \ RESET POLY /10
 5.         10 /  DUP DUP  ABS /  SWAP  1
 6.         DO    /10  10  */
 7.               DUP  I -
 8.         +LOOP
 9.         MINUS  INC !
10.                ORG !
11.         ;
12.
13. 1 +BLOCK CONTINUED
14.
15.

BLOCK 1608(3110B)
 0. ( XY STORAGE DISPLAY                              JV/771010   9 OF 12 )
 1.
 2. :ORPHAN ( 'SCALE )  ORG @ -  FRAME @  RNG @
 3.                     */ )
 4.           DUP
 5.
 6. : SCALE  SWAP  Y  ( 'SCALE ) ADOPT
 7.          SWAP  X  ( 'SCALE ) ADOPT
 8.          CORNER  D@   V+ ;
 9.
10. : *DRAW  SCALE  DRAW  ;
11.
12. 1 +BLOCK CONTINUED
13.
14.
15.
```

```
BLOCK 1609(3111B)
 0. ( XY STORAGE DISPLAY                                  JV/771010  10 OF 12 )
 1.
 2. : TICK     10 DUP  X DASH D!  ;                  \ TICKS ALONG AXES
 3.
 4. : FULL     X FRAME D@  DASH D! ;                 \ FULL LINES FOR AXES
 5.
 6. : STEP     INC @ ;
 7.
 8. : RANGE    ORG @  INC @  OVER  RNG @  +
 9.            OVER /  SWAP * ;
10.
11. 1 +BLOCK CONTINUED
12.
13.
14.
15.

BLOCK 1610(3112B)
 0. ( XY STORAGE DISPLAY                                  JV/771010  11 OF 12 )
 1. : X-GRID  X RANGE
 2.           DO    Y ORG  @ I  OFF  *DRAW
 3.                 Y DASH @ 0  ON   +DRAW
 4.                 X STEP
 5.           +LOOP ;
 6.
 7. : Y-GRID  Y RANGE
 8.           DO    X I ORG  @  OFF  *DRAW
 9.                   0 DASH @  ON   +DRAW
10.                 Y STEP
11.           +LOOP ;
12.
13. : GRID    X-GRID  Y-GRID  ;
14.
15. 1 +BLOCK CONTINUED

BLOCK 1611(3113B)
 0. ( XY STORAGE DISPLAY                                  JV/771010  12 OF 12 )
 1.
 2. : BOX    X CORNER D@  OFF  DRAW
 3.          ON  FRAME D@  SWAP
 4.          OVER 0            +DRAW
 5.          0 OVER            +DRAW
 6.          SWAP MINUS 0      +DRAW
 7.          0 SWAP MINUS      +DRAW
 8.          ;
 9.
10. FORTH DEFINITIONS
11. BASE !
12.
13. CR ." STORAGE-DISPLAY VOCABULARY AVAILABLE "
14. ;S
15.
```

```
BLOCK 1620(3124B)
 0. ( UTILITY SAVE/RESTORE                          JV/780403   1 OF ** )
 1. BASE ! OCTAL
 2. DISC-DRIVER  1000 ULP !                        \ LOAD DISC DRIVER
 3. DISC
 4. : CLU ERASE-CORE 1000 ULP ! BLOCK DROP 0 ULP ! UPDATE FLUSH ;
 5. : CUL ERASE-CORE 0 ULP ! BLOCK DROP 1000 ULP ! UPDATE FLUSH ;
 6. DECIMAL
 7. : SAVE-FIXED-PLATTER    2437 1 DO I CLU LOOP ERASE-CORE ;
 8. : RESTORE-FIXED-PLATTER 2437 1 DO I CUL LOOP ERASE-CORE ;
 9. OCTAL
10. : SAVE-FIXED-PLATTER    SAVE-FIXED-PLATTER    1000 ULP ! ;
11. : RESTORE-FIXED-PLATTER RESTORE-FIXED-PLATTER 1000 ULP ! ;
12. BASE ! ;S
13.
14.
15.

BLOCK 1630(3136B)
 0. ( UTILITY GENERATE STERREWACHT SYSTEM     JV/780403   1 OF ** )
 1. BASE !
 2. METAFORTH
 3. STERREWACHT
 4. 5 SET SAVE-AREA
 5. GENERATION
 6. SAVE
 7. FORGET METAFORTH
 8. METAFORTH
 9. STERREWACHT
10. 1 SET MASS-STORAGE
11. GENERATION
12. SAVE
13. FORGET METAFORTH
14. BASE ! ;S
15.
```

```
HP METAFORTH

PLEASE DEFINE NON-STANDARD GENERATION CONSTANTS
THEN TYPE 'GENERATION' TO GENERATE NEW SYSTEM
NEW SYSTEM WILL BE WRITTEN TO MASS STORAGE DEVICE WITH
'SAVE'

NEW SYSTEM WILL BE EXECUTED WITH 'TRY'



METAFORTH GENERATION CONSTANTS

SYSTEM SAVING BLOCKS        11 THRU 14
CHARACTERS IN ENTRY         3
MEMORY SIZE                 60000
RETURN STACK LENGTH         50
ORIGIN OF METAFORTH CODE    100
LOADER LENGTH               100
FORTH BLOCK LENGTH          1000
NUMBER OF BUFFERS           2
MAP LENGTH ( OPTIONAL )     1000
BASE PAGE CONSTANTS         200
USER AREA LENGTH 'S/R'      24
USER AREA LENGTH 'IND'      24

SYSTEM SELECT CODES
TTY CONSOLE                 14
7900 DISC SELECT CODES:12 13

COMPUTER SPECIFICATION 3
DMA AVAILABLE
HARDWARE MPY, DIV,

MASS STORAGE SPECIFICATION 3
HP 7900 DISC ,CONTINUED PLATTERS

METAFORTH BASIC GENERATION WORDS
WAIT ROUTINE 113
CONSTANT AREA STARTING AT 121
USER AREA ALLOCATION 321
INTERPRETER 371
START OF BASIC-FORTH VOCABULARY 456
CHARACTER HANDLING 1066
WORD HANDLING 1137
DICTIONARY SEARCH 1330
BLOCK HANDLING 1361
BLOCK LOADING 1711
METAFORTH INTERPRETER INPUT 2004
FORTH VOCABULARY 2014
ASSEMBLER VOCABULARY 2033
GENERAL CONSOLE I/O 2237
INTERPRETER WORDS 2304
```

```
INTERPRETER 2407
MISCELLANEOUS 2423
CONSOLE I/O 2466
MASS-STORAGE I/O 2574
HP 7900A DISC I/O 2574
FORTH BASIC STARTS 3014
HOT START 3025
COLD-START 3033
POINTER INITIALIZATION 3057
GENERATED CODE 3057
METAFORTH BINARY LENGTH 3131


END OF METAFORTH RUN
```

100	3033	55621	0	2407	0	0	0	0
110	0	2333	2333	114	0	160002	2002	24115
120	124114	177777	1	100000	2	464	3	4
130	374	377	137777	400	41000	40	177723	177720
140	56	72	6	77577	1003	57700	55672	77777
150	176775	120100	2006	121	321	200	101777	76000
160	2000	77	25	24	2311	177	120000	140000
170	10	15	10000	20000	30000	120012	4604	1000
200	177000	102004	106706	100	1056	114204	3031	0
210	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0
230	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0
310	0	0	0	0	0	0	0	0
320	0	0	321	0	0	0	55671	55622
330	55672	3131	0	2022	2022	0	0	0
340	0	0	3057	0	10	2506	2561	2023
350	0	0	0	0	0	0	0	0
360	0	0	0	0	0	0	0	0
370	0	1514	44524	0	375	160326	160000	134326
400	64101	74102	44121	74101	170101	164326	164001	134326
410	160001	124000	413	64105	44122	160001	10123	44124
420	2021	50332	24410	174331	34331	60331	3000	40101
430	2021	24405	60122	124112	34101	34102	60101	34101
440	34102	3000	40327	2021	24405	60124	124112	34101
450	34102	34101	34102	170101	60101	24441	4501	151523
460	371	2030	1600	1600	2232	2020	0	60125
470	70334	60335	170333	2400	70332	6004	24423	473
500	120000	456	471	515	2473	141517	477	467
510	511	64335	44126	160326	170001	34326	24405	520
520	60331	70335	164333	170333	40126	70331	40121	174000
530	40124	64000	24423	534	60123	70332	60333	70334
540	64335	44124	160001	170333	24405	472	20000	504
550	557	374	40	2007	570	533	510	60326
560	40121	70326	6004	174326	24406	2505	47124	545
570	557	517	503	645	454	20000	565	600
600	160101	170331	34331	24436	605	60105	40127	64332
610	6003	24400	64130	174331	34331	64000	24423	3514
620	44524	574	623	60105	64343	6020	24607	64332
630	6002	24640	64101	44121	74101	170101	60104	24400
640	64573	174331	34331	64104	24613	646	164326	160001
650	70104	134326	6004	160001	134326	24632	2462	42122
660	617	434	2104	51117	656	436	500	20000
670	662	672	160101	160000	24404	441	20000	666
700	701	160102	164101	170001	24434	453	20000	675
710	711	160101	140102	24451	1061	25440	705	720
720	134101	24405	24405	2123	53501	714	727	160102
730	164101	174102	24404	2117	53105	723	737	160102
740	24400	1504	52520	733	745	160101	24400	4103
750	47516	741	557	374	40	2007	570	577
760	510	6004	160001	24400	4126	40522	747	557
770	752	510	6004	60001	24400	2503	47525	764
1000	1001	164101	160001	1727	10131	4022	6004	174101
1010	24400	2116	52514	775	405	4101	46114	1011
1020	1021	160101	164000	54002	25033	6003	25031	114113

1030	25021	64002	174000	24436	3522	42514	1015	1040
1040	160101	164000	54002	2001	25047	6400	174000	24436
1050	1504	46501	1034	772	0	0	0	71055
1060	161054	10132	171054	61055	106706	125056	0	60337
1070	65	160000	2041	1727	10131	34337	125066	1102
1100	40640	1050	1103	160101	70337	15066	24404	0
1110	64336	4065	74106	34336	2041	1727	64000	60131
1120	2040	1727	110106	30001	170106	125107	1102	20440
1130	1077	1132	160101	70336	160102	15107	24434	1140
1140	164101	4022	60340	74340	40340	70337	64331	2400
1150	170001	6004	170001	4022	44121	74336	15066	150102
1160	25156	15107	60133	140331	170331	3004	40134	2020
1170	25175	15066	150102	2001	25161	60135	15107	60340
1200	3004	40337	70340	24434	1205	60341	2002	24400
1210	24377	3116	52515	1126	1215	60331	22	2004
1220	70337	2400	70104	70105	20123	70343	15066	40136
1230	70110	2003	25236	60337	40121	70337	60344	70107
1240	15066	70106	40137	2020	25276	3004	40107	3004
1250	2021	25276	40107	70106	60104	100200	107	70104
1260	60105	100200	107	6020	44107	2100	40106	2040
1270	6004	44104	74104	70105	34343	25236	60106	50140
1300	25325	50141	25323	50135	25310	134326	134326	24405
1310	60110	2002	24405	64104	60105	7000	3007	6004
1320	74104	70105	24405	60142	25237	2400	70343	25236
1330	2106	44516	1211	1334	60331	2004	160000	70104
1340	164331	160334	154000	25351	40124	160000	2002	25342
1350	25305	70105	2004	160000	10143	50104	24405	60105
1360	25344	4502	46117	1330	557	665	503	5102
1370	46117	1361	557	2743	503	5502	46117	1367
1400	557	2757	503	1507	42524	1375	557	374
1410	3	1020	503	1524	42507	1403	557	374
1420	3	1037	503	2120	51105	1413	772	55672
1430	56675	557	1426	671	717	736	1372	1426
1440	671	726	736	700	717	744	374	1000
1450	710	374	20076	736	700	374	41040	726
1460	717	700	503	557	1426	671	717	1426
1470	671	671	1400	503	1475	61430	64000	40144
1500	50145	60146	71430	55427	25475	75427	160001	2020
1510	25513	34326	24405	10147	170001	24405	1517	161427
1520	10147	150101	25535	64145	44150	160001	10147	150101
1530	25534	54146	24405	25524	75427	34326	61427	2004
1540	24404	5105	51101	1423	1545	64145	2400	44150
1550	170001	54146	24405	25547	3125	50104	1541	1560
1560	161427	30123	171427	24405	557	1474	1463	503
1570	3101	41114	1554	557	1406	1516	1564	1431
1600	503	2502	46117	1570	557	1014	1573	503
1610	557	1516	1564	1426	671	726	736	700
1620	717	503	3102	525006	1601	557	1014	1406
1630	1610	1557	503	4122	425001	1622	557	1406
1640	1564	1426	671	717	744	1364	374	0
1650	726	700	503	2506	46125	1633	557	1406
1660	1564	1564	503	1077	52440	1653	1667	160101
1670	40151	2021	24405	40152	2020	24405	6400	100400
1700	144	100200	144	40146	64000	60123	130001	170001
1710	24405	2114	47501	1663	1715	60326	40121	64340
1720	174000	40121	64341	174000	40121	70326	6400	74340
1730	164101	74341	64103	174326	24436	1073	51440	1711
1740	1741	34326	164326	74341	34326	164326	74340	24515
1750	4503	47516	1735	1754	61430	65427	75430	71427
1760	25726	1076	141040	1750	1765	64101	74102	44121

1770	74101	60341	2004	170101	25754	3053	41114	1761
2000	2001	60341	140101	24404	2127	47522	1775	557
2010	1204	1604	1137	503	2506	147522	2004	2025
2020	1600	1600	3014	0	465	6004	74465	44121
2030	44126	74334	24405	207	0	164101	60153	154000
2040	126034	2004	52033	2001	26037	50154	26052	174000
2050	36033	126034	60126	124112	443	20000	463	2060
2060	16034	24404	2063	160101	1000	1727	24404	2070
2070	160101	1323	30155	26065	1115	51040	2054	557
2100	2062	752	510	160101	10156	150101	26124	160101
2110	20331	10157	2002	26120	160101	10156	30160	26124
2120	74105	16034	64105	30123	6004	140001	24601	1503
2130	50125	2074	557	752	510	6004	160001	170331
2140	34331	24405	2104	42501	2127	557	2067	752
2150	510	6004	160001	170331	34331	24600	1511	27517
2160	2142	557	2067	752	510	160101	10161	26124
2170	451	20000	2156	2174	160101	30123	24404	2105
2200	42117	2170	2203	60333	70334	24405	2516	42530
2210	2177	2135	24405	2462	50117	2206	2135	24434
2220	3101	41117	2213	2135	124111	3105	51122	2220
2230	2135	124112	3120	40525	2225	2135	114113	2124
2240	54520	2014	2243	64162	44002	164001	160101	2002
2250	24410	24434	3105	54120	2237	2256	64101	74102
2260	44121	74101	160102	170101	60342	40000	170102	64163
2270	44002	164001	24410	2121	52511	2252	557	1416
2300	374	120	2255	503	2305	60103	170326	24405
2310	477	1416	1000	2242	374	2310	1000	2242
2320	374	120	2255	2304	2517	45415	3412	2501
2330	41117	2273	2333	60331	64327	74101	64330	44121
2340	74326	6400	74341	74332	64164	174326	24400	2505
2350	51122	2327	2353	160101	124112	400	120000	2347
2360	557	1416	374	2324	1000	2242	374	120
2370	2255	503	447	120000	2355	557	374	40
2400	2007	1333	604	503	374	17	2352	374
2410	40	2007	1333	412	2304	1214	622	2304
2420	374	16	2352	2103	47504	2372	557	374
2430	40	2007	570	461	503	450	120000	2423
2440	557	374	51	2007	503	5504	42506	2435
2450	2451	60334	70333	24405	3123	54523	2445	761
2460	321	3	3	3	24	24	3	106614
2470	102614	103714	114113	102314	26472	103514	10165	126466
2500	0	64166	16466	50165	126500	124111	2507	160102
2510	164101	44000	174101	70336	60167	16466	2003	124111
2520	70106	50170	26550	50171	26535	16500	60106	15107
2530	160101	3004	40336	2020	26514	60135	16500	60135
2540	15107	2400	15107	60135	15107	160102	70340	24434
2550	60170	16500	160102	3000	40336	2020	26514	140102
2560	26513	2562	164101	7004	174101	160102	70337	15066
2570	16500	134101	26567	24434	0	1000	0	0
2600	2601	160101	40121	6400	100400	176	170101	60001
2610	6400	100400	142	65	72576	2400	1600	1727
2620	5222	30001	72577	62575	164101	6002	3000	10177
2630	42577	72577	24436	0	46574	106613	103713	102312
2640	26637	126633	0	102313	26643	102106	106712	106713
2650	6400	16633	106512	4010	102044	126642	0	106712
2660	106713	66576	106612	103712	64174	16633	66577	106612
2670	103712	16642	126656	0	16656	106712	106713	64175
2700	106606	106702	130101	102602	102702	642000	106602	126673
2710	2711	60123	16673	60173	32574	102613	103712	103706
2720	103713	24436	2723	2400	16673	102712	102112	103706

2730	60172	32574	102613	103713	24436	2736	16642	24405
2740	2504	51105	2454	557	1053	1020	2600	2710
2750	2735	1053	1037	503	3104	53522	2740	557
2760	1053	1020	2600	2722	2735	1053	1037	503
2770	2104	44523	2754	557	374	665	374	1365
3000	700	374	2743	374	1373	700	374	2757
3010	374	1401	700	503	1107	47440	2770	557
3020	374	15	1714	503	3025	62460	70002	62363
3030	26334	1544	3024	60201	70004	2400	70005	70003
3040	106700	102100	60202	64142	170001	6004	2004	54203
3050	2001	27044	60205	70006	60206	170326	24405	0

```
HP METAFORTH

PLEASE DEFINE NON-STANDARD GENERATION CONSTANTS
THEN TYPE 'GENERATION' TO GENERATE NEW SYSTEM
NEW SYSTEM WILL BE WRITTEN TO MASS STORAGE DEVICE WITH
'SAVE'

NEW SYSTEM WILL BE EXECUTED WITH 'TRY'


METAFORTH GENERATION CONSTANTS

SYSTEM SAVING BLOCKS          11 THRU 14
CHARACTERS IN ENTRY           FLOATING
MEMORY SIZE                   60000
RETURN STACK LENGTH           50
ORIGIN OF METAFORTH CODE      100
LOADER LENGTH                 100
FORTH BLOCK LENGTH            1000
NUMBER OF BUFFERS             2
MAP LENGTH ( OPTIONAL )       1000
BASE PAGE CONSTANTS           200
USER AREA LENGTH 'S/R'        24
USER AREA LENGTH 'IND'        24

SYSTEM SELECT CODES
TTY CONSOLE              14
7900 DISC SELECT CODES:12 13

COMPUTER SPECIFICATION 3
DMA AVAILABLE
HARDWARE MPY, DIV,

MASS STORAGE SPECIFICATION 3
HP 7900 DISC ,CONTINUED PLATTERS

METAFORTH BASIC GENERATION WORDS
WAIT ROUTINE 113
CONSTANT AREA STARTING AT 121
USER AREA ALLOCATION 321
INTERPRETER 371
START OF BASIC-FORTH VOCABULARY 464
CHARACTER HANDLING 1160
WORD HANDLING 1231
DICTIONARY SEARCH 1424
BLOCK HANDLING 1503
BLOCK LOADING 2066
METAFORTH INTERPRETER INPUT 2167
FORTH VOCABULARY 2200
ASSEMBLER VOCABULARY 2220
GENERAL CONSOLE I/O 2436
INTERPRETER WORDS 2507
```

```
INTERPRETER 2614
MISCELLANEOUS 2630
CONSOLE I/O 2702
MASS-STORAGE I/O 3010
HP 7900A DISC I/O 3010
FORTH BASIC STARTS 3234
HOT START 3245
COLD-START 3253
POINTER INITTALIZATION 3277
GENERATED CODE 3277
METAFORTH BINARY LENGTH 3351


END OF METAFORTH RUN
```

100	3253	55621	0	2614	0	0	0	0
110	0	2537	2537	114	0	160002	2002	24115
120	124114	177777	377	100000	2	1	475	3
130	374	137777	400	41000	40	177723	177720	56
140	72	6	77577	1003	57700	55672	77777	176775
150	120100	2006	121	321	200	101777	76000	2000
160	77	25	24	2514	177	120000	140000	10
170	15	10000	20000	30000	120012	4604	1000	177000
200	102004	106706	100	1150	114203	3251	0	0
210	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0
230	0	0	0	0	0	0	0	0
240	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0
310	0	0	0	0	0	0	0	0
320	0	0	321	0	0	0	55671	55622
330	55672	3351	0	2207	2207	0	0	0
340	0	0	3277	0	10	2722	2775	2210
350	0	0	0	0	0	0	0	0
360	0	0	0	0	0	0	0	0
370	0	1514	44524	0	375	160326	160000	134326
400	64101	74102	44121	74101	170101	164326	164001	134326
410	160001	124000	413	64105	160001	1727	10122	21
420	2003	2004	44000	160001	10123	44124	2021	50332
430	24410	174331	34331	60331	3000	40101	2021	24405
440	60125	124112	34101	34102	60101	34101	34102	3000
450	40327	2021	24405	60124	124112	34101	34102	34101
460	34102	170101	60101	24447	4501	51523	42515	41114
470	142522	371	2215	1600	1600	2427	2205	0
500	60126	70334	60335	170333	2400	70332	6004	24431
510	473	120000	464	502	536	2473	41517	142105
520	510	500	523	64335	160001	1727	10122	21
530	2003	2004	40124	44000	160326	170001	34326	24405
540	541	60331	70335	164333	170333	160000	1727	10122
550	21	2003	2004	40124	40331	70331	40121	174000
560	40124	64000	24431	564	60123	70332	60333	70334
570	64335	160001	1727	10122	21	2003	2004	2004
600	44000	160001	170333	24405	472	20000	515	616
610	374	40	2173	630	563	522	60326	40121
620	70326	6004	174326	24406	2505	47124	42522	604
630	616	540	514	716	454	20000	624	640
640	160101	170331	34331	24444	645	60105	160000	1727
650	10122	21	2003	2004	40127	40105	64332	6003
660	24400	64130	174331	34331	64000	24431	3514	44524
670	42522	40514	634	674	60105	64343	6020	24656
700	64332	6002	24711	64101	44121	74101	170101	60104
710	24400	64633	174331	34331	64104	24662	717	164326
720	160001	70104	134326	6004	160001	134326	24703	2462
730	42122	47520	666	442	2104	51117	50040	727
740	444	500	20000	734	745	160101	160000	24404
750	441	20000	741	754	160102	164101	170001	24442
760	453	20000	750	764	160101	140102	24457	1061
770	25440	760	773	134101	24405	24405	2123	53501
1000	50040	767	1003	160102	164101	174102	24404	2117
1010	53105	51040	776	1014	160102	24400	1504	52520
1020	1007	1022	160101	24400	4103	47516	51524	40516

	1030	52040	1016	615	374	40	2173	630	637
2	1040	522	6004	160001	24400	4126	40522	44501	41114
	1050	42440	1024	616	1032	522	6004	60001	24400
4	1060	2503	47525	47124	1044	1065	164101	160001	1727
	1070	10122	4022	6004	174101	24400	2116	52514	46040
6	1100	1060	405	4101	46114	47503	40524	42440	1075
	1110	1111	160101	164000	54002	25123	6003	25121	114113
8	1120	25111	64002	174000	24444	3522	42514	42501	51505
	1130	1102	1132	160101	164000	54002	2001	25141	6400
10	1140	174000	24444	1504	46501	1124	1055	0	0
	1150	0	71147	161146	10131	171146	61147	106706	125150
12	1160	0	60337	65	160000	2041	1727	10122	34337
	1170	125160	1102	40040	1142	1175	160101	70337	15160
14	1200	24404	0	64336	4065	74106	34336	2041	1727
	1210	64000	60122	2040	1727	110106	30001	170106	125201
16	1220	1102	20440	1171	1224	160101	70336	160102	152001
	1230	24442	1232	164101	4022	60340	74340	40340	70337
18	1240	64331	2400	170001	6004	170001	4022	44121	74336
	1250	15160	150102	25250	15201	60132	140331	170331	3004
20	1260	40133	2020	25267	15160	150102	2001	25253	60134
	1270	15201	60340	3004	40337	70340	24442	1277	60341
22	1300	2002	24400	24377	3116	52515	41105	51040	1220
	1310	1311	60331	22	2004	70337	2400	70104	70105
24	1320	20123	70343	15160	40135	70110	2003	25332	60337
	1330	40121	70337	60344	70107	15160	70106	40136	2020
26	1340	25372	3004	40107	3004	2021	25372	40107	70106
	1350	60104	100200	107	70104	60105	100200	107	6020
28	1360	44107	2100	40106	2040	6004	44104	74104	70105
	1370	34343	25332	60106	50137	25421	50140	25417	50134
30	1400	25404	134326	134326	24405	60110	2002	24405	64104
	1410	60105	7000	3007	6004	74104	70105	24405	60141
32	1420	25333	2400	70343	25332	2106	44516	42040	1303
	1430	1431	160331	1727	10122	21	2003	2004	3004
34	1440	70107	160331	70110	164334	160001	50110	25462	1727
	1450	10122	21	2003	2004	2004	44000	164001	6002
36	1460	25444	25401	74105	60107	70104	60331	70106	34106
	1470	6004	160001	10142	150106	25500	64105	160001	25447
38	1500	34104	25467	24405	4502	46117	41513	26515	40520
	1510	1424	616	740	514	5102	46117	41513	26522
40	1520	425001	42040	1503	616	3160	514	5502	46117
	1530	41513	26527	51111	52105	1514	616	3176	514
42	1540	1507	42524	1526	616	374	3	1110	514
	1550	1524	42507	1540	616	374	3	1131	514
44	1560	2120	51105	53040	1550	1055	55672	56675	616
	1570	1564	744	772	1013	1523	1564	744	1002
46	1600	1013	753	772	1021	374	1000	763	374
	1610	20076	1013	753	374	41040	1002	772	753
48	1620	514	616	1564	744	772	1564	744	744
	1630	1535	514	1633	61566	64000	40143	50144	60145
50	1640	71566	55565	25633	75565	160001	2020	25651	34326
	1650	24405	10146	170001	24405	1655	161565	10146	150101
52	1660	25673	64144	44147	160001	10146	150101	25672	54145
	1670	24405	25662	75565	34326	61565	2004	24404	5105
54	1700	51101	51505	26503	47522	42440	1560	1707	64144
	1710	2400	44147	170001	54145	24405	25711	3125	50104
56	1720	40524	42440	1677	1724	161565	30123	171565	24405
	1730	616	1632	1621	514	3101	41114	47503	45440
58	1740	1716	616	1543	1654	1730	1567	514	2502
	1750	46117	41513	1734	616	1101	1741	514	616
60	1760	1654	1730	1564	744	1002	1013	753	772
62									
64									

1770	514	3102	52506	43105	51040	1747	616	1101	
2000	1543	1757	1723	514	4122	42501	42055	46501	
2010	50040	1771	616	1543	1730	1564	744	772	
2020	1021	1511	374	0	1002	753	514	2506	
2030	46125	51510	2004	616	1543	1730	1730	514	
2040	1077	52440	2027	2044	160101	40150	2021	24405	
2050	40151	2020	24405	6400	100400	143	100200	143	
2060	40145	64000	60123	130001	170001	24405	2114	47501	
2070	42040	2040	2073	60326	40121	64340	174000	40121	
2100	64341	174000	40121	70326	6400	74340	164101	74341	
2110	64103	174326	24444	1073	51440	2066	2117	34326	
2120	164326	74341	34326	164326	74340	24536	4503	47516	
2130	52111	47125	42504	2113	2135	61566	65565	75566	
2140	71565	26104	1076	141040	2126	2146	64101	74102	
2150	44121	74101	60341	2004	170101	26135	3053	41114	
2160	47503	45440	2142	2164	60341	140101	24404	2127	
2170	47522	42040	2156	616	1276	1753	1231	514	
2200	2506	47522	152110	2167	2212	1600	1600	3234	
2210	0	476	6004	74476	44121	44127	74334	24405	
2220	206	0	164101	60152	154000	126221	2004	52220	
2230	2001	26224	50153	26237	174000	36220	126221	60127	
2240	124112	443	20000	474	2245	16221	24404	2250	
2250	160101	1000	1727	24404	2255	160101	1323	30154	
2260	26252	1115	51040	2241	616	2247	1032	522	
2270	160101	10155	150101	26311	160101	20331	10156	2002	
2300	26305	160101	10155	30157	26311	74105	16221	64105	
2310	30123	6004	140001	24641	1503	50125	2261	616	
2320	1032	522	6004	160001	170331	34331	24405	2104	
2330	42501	52440	2314	616	2254	1032	522	6004	
2340	160001	170331	34331	24640	1511	27517	2327	616	
2350	2254	1032	522	160101	10160	26311	451	20000	
2360	2344	2362	160101	30123	24404	2105	42117	41440	
2370	2356	2372	60333	70334	24405	2516	42530	52054	
2400	2365	2322	24405	2462	50117	50054	2375	2322	
2410	24442	3101	41117	51124	26040	2403	2322	124111	
2420	3105	51122	47522	26040	2411	2322	124112	3120	
2430	40525	51505	26040	2420	2322	114113	2124	54520	
2440	42440	2200	2443	64161	44002	164001	160101	2002	
2450	24410	24442	3105	54120	42503	52040	2436	2460	
2460	64101	74102	44121	74101	160102	170101	60342	40000	
2470	170102	64162	44002	164001	24410	2121	52511	52040	
2500	2452	616	1553	374	120	2457	514	2510	
2510	60103	170326	24405	477	1553	1064	2442	374	
2520	2513	1064	2442	374	120	2457	2507	2517	
2530	45415	3412	2501	41117	51124	2475	2537	60331	
2540	64327	74101	64330	44121	74326	6400	74341	74332	
2550	64163	174326	24400	2505	51122	47522	2532	2560	
2560	160101	124112	400	120000	2553	616	1553	374	
2570	2527	1064	2442	374	120	2457	514	447	
2600	120000	2562	616	374	40	2173	1430	644	
2610	514	374	17	2557	374	40	2173	1430	
2620	412	2507	1310	673	2507	374	16	2557	
2630	2103	47504	42440	2577	616	374	40	2173	
2640	630	472	514	450	120000	2630	616	374	
2650	51	2173	514	5504	42506	44516	44524	44517	
2660	47123	2643	2663	60334	70333	24405	3123	54523	
2670	52105	46440	2653	1041	321	0	3	3	
2700	24	24	0	106614	102614	103714	114113	102314	
2710	26706	103514	10164	126702	0	64165	16702	50164	
2720	126714	124111	2723	160102	164101	44002	174101	70336	

2730	60166	16702	2003	124111	70106	50167	26764	50470	
2740	26751	16714	60106	15201	160101	3004	40336	2020	
2750	26730	60134	16714	60134	15201	2400	15201	60134	
2760	15201	160102	70340	24442	60167	16714	160102	3000	
2770	40336	2020	26730	140102	26727	2776	164101	7004	
3000	174101	160102	70337	15160	16714	134101	27003	24442	
3010	0	1000	0	0	3015	160101	40121	6400	
3020	100400	175	170101	60001	6400	100400	141	65	
3030	73012	2400	1600	1727	5222	30001	73013	63011	
3040	164101	6002	3000	10176	43013	73013	24444	0	
3050	47010	106613	103713	102312	27053	127047	0	102313	
3060	27057	102106	106712	106713	6400	17047	106512	4010	
3070	102044	127056	0	106712	106713	67012	106612	103712	
3100	64173	17047	67013	106612	103712	17056	127072	0	
3110	17072	106712	106713	64174	106606	106702	130101	102602	
3120	102702	64177	106602	127107	3125	60123	17107	60172	
3130	33010	102613	103712	103706	103713	24444	3137	2400	
3140	17107	102712	102112	103706	60171	33010	102613	103713	
3150	24444	3152	17056	24405	2504	51105	40504	2666	
3160	616	1145	1110	3014	3124	3151	1145	1131	
3170	514	3104	53522	44524	42440	3154	616	1145	
3200	1110	3014	3136	3151	1145	1131	514	2104	
3210	44523	41440	3171	616	374	740	374	1512	
3220	753	374	3160	374	1524	753	374	3176	
3230	374	1536	753	514	1107	47440	3207	616	
3240	374	15	2072	514	3245	62674	70002	62570	
3250	26540	1706	3244	60200	70004	2400	70005	70003	
3260	106700	102100	60201	64141	170001	6004	2004	54202	
3270	2001	27264	60204	70006	60205	170326	24405	0	

2730	60166	16702	2003	124111	70106	50167	26764	50170	
2740	26751	16714	60106	15201	160101	3004	40336	2020	
2750	26730	60134	16714	60134	15201	2400	15201	60134	
2760	15201	160102	70340	24442	60167	16714	160102	3000	
2770	40336	2020	26730	140102	26727	2776	164101	7004	
3000	174101	160102	70337	15160	16714	134101	27003	24442	
3010	0	1000	0	0	3015	160101	40121	6400	
3020	100400	175	170101	60001	6400	100400	141	65	
3030	73012	2400	1600	1727	5222	30001	73013	63011	
3040	164101	6002	3000	10176	43013	73013	24444	0	
3050	47010	106613	103713	102312	27053	127047	0	102313	
3060	27057	102106	106712	106713	6400	17047	106512	4010	
3070	102044	127056	0	106712	106713	67012	106612	103712	
3100	64173	17047	67013	106612	103712	17056	127072	0	
3110	17072	106712	106713	64174	106606	106702	130101	102602	
3120	102702	64177	106602	127107	3125	60123	17107	60172	
3130	33010	102613	103712	103706	103713	24444	3137	2400	
3140	17107	102712	102112	103706	60171	33010	102613	103713	
3150	24444	3152	17056	24405	2504	51105	40504	2666	
3160	616	1145	1110	3014	3124	3151	1145	1131	
3170	514	3104	53522	44524	42440	3154	616	1145	
3200	1110	3014	3136	3151	1145	1131	514	2104	
3210	44523	41440	3171	616	374	740	374	1512	
3220	753	374	3160	374	1524	753	374	3176	
3230	374	1536	753	514	1107	47440	3207	616	
3240	374	15	2072	514	3245	62674	70002	62570	
3250	26540	1706	3244	60200	70004	2400	70005	70003	
3260	106700	102100	60201	64141	170001	6004	2004	54202	
3270	2001	27264	60204	70006	60205	170326	24405	0	

3

5' 5@

: ^ 3DUP ^ 3 ^ PICK ^ 3 ^ PICK ^ 3 ^ PICK ^ ;

1624

```
@OK
?GO


    YOK
?2 3 + . 5 OK
2 HERE PLUMP OK


HERE . 4123 OK

( DIT IS FORTH ) OK

HERE 1 + . 4124 OK
5 2 ↑ . ↑? OK
5 2 ** . **? OK
7 4 * . 28 OK

1 2 3 4 5 6 7 * + * + * + . 383 OK
4 . 4 OK
3 4 * 1 2 + + 9 3 / / 7 8 * - . -51 OK
1 2 3 4 5  OK
DUP OK
. . . . . 5 5 4 3 2 OK
. 1 OK
1 2 3 4 5 6 7 2DUP 2DUP? OK
: 2DUP ; OK
FORGET 2DUP OK
. -29236.?

          OK
1 2 3 4 5 6 7  OK
. 7 OK
ROT OK
. . . . . 4 6 5 3 2 OK
```

DUP .

: OKTO OCTAL . DECIMAL ;

1234 OKTO.

10

BASE @ BASE.

3

5' SG

: 3DUP 3 PICK 3 PICK 3 PICK ;

```
@OK
?GO

    YOK
?2 3 + . 5 OK
2 HERE PLUMP OK

HERE . 4123 OK

( DIT IS FORTH ) OK

HERE 1 + . 4124 OK
5 2 ↑ . ↑? OK
5 2 ** . **? OK
7 4 * . 28 OK

1 2 3 4 5 6 7 * + * + * + . 383 OK
4 . 4 OK
3 4 * 1 2 + + 9 3 / / 7 8 * - . -51 OK
1 2 3 4 5  OK
DUP OK
. . . . .  5 5 4 3 2 OK
. 1 OK
1 2 3 4 5 6 7 2DUP 2DUP? OK
: 2DUP ; OK
FORGET 2DUP OK
. -29236.?

          OK
1 2 3 4 5 6 7  OK
. 7 OK
ROT OK
. . . . . 4 6 5 3 2 OK
```

: OKTO ' OCTAL . DECIMAL ;

1234 OKTO.

10

BASE in BASE.

HALT

uit

Sleutel → standby.

Disk schakelaar ↓

Terminal af.

Komp jacks achter af.

aan

Terminals aan.

Disk → load.

Komp j. aan.

↑ R ↓

operate.

↑

→ 5.

clear display

register

store.

zie plakkertje

go op Tracer.

1631 LOAD.

go op HP.

~~2437~~

2437 LOAD